NATIONAL INSTITUTE OF AEROSPACE

Technical Report for

FAA # DTFAWA-09-A-8000

APA Suite Sensitivity Study & Model Scoring Methodologies

Prepared by:

Karen Feigh, Lakshmi N. Sankar, and Daniel P. Schrage

Brian Huff, Ashley Flick and Vasu Manivannan

Task Monitor:

Fred Brooks

TABLE OF CONTENTS

Technical Report
Task Number: DTFAWA-09-A-8000

Date:	**December 29, 2010**
Title:	**APA Suite Sensitivity Study & Model Scoring Methodologies**
PI Name & Affiliation:	**Dr. Karen Feigh, Dr. Lakshmi Sankar, Dr. Dan Schrage, Brian Huff, Ashley Flick and Vasu Manivannan** **Georgia Institute of Technology**
Task Monitor:	**Fred Brooks**

EXECUTIVE SUMMARY

The primary objective of the present study is to systematically evaluate the current real-time wake vortex modeling methods. Specifically, the present study evaluates existing real-time wake vortex computational models against measured wake vortex data provided by the sponsor. As an initial step, a parametric study is done to determine the meteorological data resolution needed for accurate real-time modeling of wake vortex behavior from the surface to an altitude of 60,000 feet.

For final approach and initial climb flight segments, previously collected meteorological and wake vortex data are analyzed to identify specific data sets suitable for a detailed scoring of existing real-time wake vortex computational models. Metrics are established to assess the meteorological and wake vortex data quality. Data meeting these metrics are used to perform sensitivity and accuracy assessments on the computational wake models. The results from this assessment, including the sensitivity assessment, will be used to develop concepts of operation for avoiding aircraft wake vortices and to improve deterministic wake prediction models.

This report documents the findings of this research study.

Task 1: Examination of Existing Real-Time Wake Data

An initial parametric study of the meteorological data resolution needed for accurate real-time modeling of wake vortex behavior from the surface to an altitude of 60,000 feet is conducted. In the lower atmosphere, where previously collected data sets of meteorological and wake data exist, a set of verification criteria is being established to determine the suitability of these data sets for wake vortex modeling scoring activities. Using the established criteria, data sets provided by the sponsor are being analyzed and their suitability determined. Data sets which meet the requirements are assembled into a single data set or repository from which researchers can draw specific cases for wake modeling scoring analyses and model validation activities.

Task 1 has been divided into the following sub-tasks.

<u>Task 1.1</u>: Parametric study of meteorological and wake data resolutions between 60,000 feet and the surface to determine the data resolution needed in height and time for accurate real-time wake modeling.

<u>Task 1.2</u>: Establish a set of qualitative and quantitative metrics to evaluate the current and future real-time meteorological and wake vortex data sets.

<u>Task 1.3</u>: Evaluation of available meteorological and wake vortex data sets collected in the lower atmosphere using the previously identified qualitative and quantitative metrics.

<u>Task 1.4</u>: Using the outcome of the evaluation, identification of data sets. These data sets will be used in Tasks 2 and 3 to develop methodologies for scoring the performance of real-time wake models and to conduct model scoring studies.

<u>Task 1.5</u>: Identification of any missing data necessary or beneficial for future analysis of real-time, computational, wake vortex models.

Task 2: Development of Scoring Methodologies and Tools

The objective of Task 2 is to develop consistent scoring methodologies to compare the performance of the both deterministic and probabilistic real-time wake vortex models. The following activities are carried out:

Task 2.1: A review on existing real time wake vortex prediction models and quantify the assumptions within the models.

Task 2.2: Establish scoring methodology for the wake vortex prediction models against test data. At present, researchers use the RMS values of the differences between predicted and measured vortex strength, the differences between predicted and measured vortex (y,z) positions, and differences between time elapsed before the vortex leaves the corridor. Researchers also use 90 percentile errors to score the models. Additional metrics are developed for scoring the data. Rationales for when to accept or reject test data which appear to be outliers are also developed. Automated tools are developed for performing the actual scoring, starting with the tools that are already available in the public domain.

Task 3: Wake Vortex Model Scoring

In this task, the methodologies, tools, and rationales developed in task 2 are used to score the wake vortex models against data identified in Task 1. At each site, the results are grouped by aircraft size and model type. In the past, only a limited set of data (Memphis, Dallas Fort Worth, and Denver) has been used using a limited number of models. The current study performs a more comprehensive analysis using all the models available to the researchers, and all the data that has been certified under Task 1, using the scoring methodologies developed under Task 2.

TASK 1: APA SUITE SENSITIVITY STUDY

OVERVIEW OF THE APA SUITE AND AVAILABLE DATA

APA SUITE

Software for the APA Suite was provided to Georgia Tech as an executable file with sample input and output files for reference along with a User's Guide. The provided executable file operates in an external mode which enables the use of two different wake vortex fast-time models: the APA Suite Prediction Algorithm (APA) 3.2.2 and TASS Driven Algorithms for Wake Prediction (TDAWP).

The APA model predicts wake vortex trajectories and circulation on a plane perpendicular to the path of the generating aircraft. The initial wake is represented as two vortices whose initial strength and position are dependent upon input conditions. The decay of the vortices is computed with a formula derived from a TASS LES study. If the wake is out-of-ground effect, the APA model utilizes a decay and transport model. For in-ground effect, APA utilized image vortices to model the rebound of the wake vortex.

Execution of the APA Suite results in the generation of two output files: *APA.out* and *TRAJECT.dat*. The *APA.out* file is a summary file detailing the program execution and listing any problems encountered. The *TRAJEC.dat* file is the code's prediction of the wake vortex trajectory. The first column is time. The next three columns represent the vortex circulation, horizontal position, and vertical position. Results are available for both the starboard and port vortices. Each additional set of three columns after the seventh, represent secondary and tertiary vortices if the analysis determines the presence of ground effects.

The TDAWP model has separate prognostic equations for vortex descent rate and 5-15m average circulation, and it applies these separately to both the port and starboard vortices. The formulation is driven by parametric studies from LES using TASS. The TDAWP formulation also includes the effects of crosswind shear on vortex descent rate, thus allowing the prediction of vortex tilt and the change in lateral separation due to crosswind.

The APA Suite requires the following five data files in order to run:

- The cross-flow velocity profile (m/s) "UDATA.dat"
- The environmental turbulence (m^2/s^3) "QDATA.dat"
- The environmental temperature profile (°C or potential temperature (K)) "TDATA.dat"
- Aircraft specification file "ACDATA.dat"
- Model control file "APA.in"

The *QDATA*, *TDATA*, and *UDATA* files are saved in a two column format, where the first column is height (m) and the second column is the parameter value (Q, T, or U). The aircraft specification file contains a row of four values: the lateral and vertical position of the aircraft (m), initial vortex descent rate (m/s), and initial vortex separation (m) [1]. The initial vortex separation b_o is equivalent to the aircraft wingspan multiplied by a factor of $\pi/4$. The *APA.in* file allows the user to specify either APA for an APA 3.2.2 analysis or TDP for a TDAWP analysis.

AVAILABLE METEOROLOGICAL DATA

A total of 3 data sets were available for analysis: Memphis, TN (December 1994 and August 1995); Dallas/Fort Worth, TX (November and December 1999); and Idaho Falls, ID (June and September 1990). The data was catalogued, following an extensive investigation into the numerous data files. Specifically, each file type and a brief description were catalogued in a file named "Meteorological Data ToC.xls". Following the cataloging process, the necessary meteorological data files as well as the measured vortex data files were extracted, and two databases were created: one for meteorological files and one for measured vortex files. The meteorological database (Meteorological_Database.xls) contains the filenames for the raw data and corresponding information about the run (i.e. data, time, and weather conditions if known). Interpolated filenames were not included due to redundancy.

The weather conditions at DFW are obtained from a combination of two sources. The detailed description is found from a file titled "daily_reports.pdf" in the AVOSS_DFW99 subset of data. This file provides a detailed description of the weather during each day of data collection. In conjunction with these detailed descriptions, the basic wind speed and cloud conditions are taken from the file "wind_clouds99.pdf" in the AVOSS_DFW99 subset of the data. This file categorizes the data by the time of day the measurement is taken and sorts the data into four separate weather categories: cloudy (cover > 5/10) and windy (speed > 10 kts), cloudy and calm, clear and windy, and clear and calm. This categorization provides a detailed metric of the weather conditions that can easily be compared between the different measurement periods.

The weather conditions at MEM are obtained from pages 67-70 of NASA Contract Report 201690 [2]. A description of the meteorological conditions is provided for each time of aircraft measurement. Using the detailed description, the weather is again categorized into the four different weather categories, as described above for DFW. The weather categories along with a detailed description of the daily weather conditions are added to the meteorological database file for each airport.

The weather conditions at Idaho Falls were not available.

The vortex database (Vortex_Database.xlsx) contains the location, date, time, y-value, v_o, b_o, GT case ID number, aircraft type, and filename. Each vortex file has been given a unique GT#### case ID number for easy identification. These ID numbers are used to label files created from the MATLAB program described later.

The meteorological data from each of the three locations (DFW, Memphis, and Idaho Falls) differ from each other, and required separate methods of handling. For each location, the required UDATA, TDATA, and QDATA files were created by parsing the original data files into a standard format required by the APA Suite v4.19 program. In the absence of environmental turbulence (QDATA) for a particular measurement time, a constant profile of $1x10^{-4}$ m^2/s^3 is assumed based on private communications with Dr. Fred Proctor at NASA.

The meteorological data files provided for MEM and DFW cover an altitude range of 0-1,400m (some only 600 m). Due to the lack of data available to cover the entire 0-20,000m range, an investigation was performed to obtain a collection of standardized meteorological data sets for altitudes providing data from 0-20,000m. This standardized data is to be used as a baseline comparison for meteorological data collected from the airport measurements. Both the temperature and crosswind velocity profile were investigated. Sample high altitude meteorological data files are taken from the National Oceanic Atmospheric Administration's (NOAA) website [3] for two different days of measurement. The first data set is referred to as ATL 1 in Figure 1 and was measured on March 4th, 2009 and 12:00 AM. The file provides measurements from 244m to 26,210m. The second data set is hereby referred to as ATL 2 in Figure 1 and was measured on October 1st, 2005 at 12:00 AM. The file provides measurements from 224m to 31,200m. These files are compared with measurements from DFW provided the highest available altitude of measurement out of any of the airports. The case chosen is 991110_000000_DFW. For this case both the TDATA and UDATA files are compared to the data from Atlanta.

The temperature data collected from the various locations is shown in Figure 1. The temperature appears to follow similar behavior at each location, with the temperature converging to a similar range of values after approximately 2,000 meters. The STD_ATMOS line represents the atmospheric lapse rate of 6.4°C/km in the Troposphere and a constant change of temperature within the

Tropopause. There does not appear to be any drastic variation in the temperature profiles, at least in the region above 2,000 m. Below this altitude, numerous factors including time of day and geographic location contribute to significant variations within the temperature distributions.

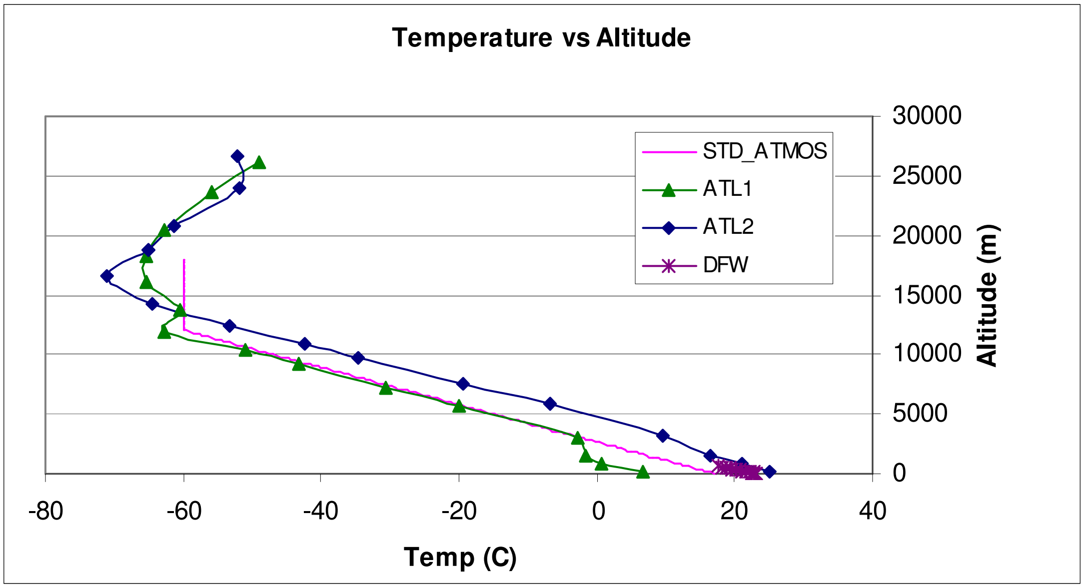


FIGURE 1: EXTRACTION AND ESTIMATES OF TEMPERATURE AS A FUNCTION OF ALTITUDE

There is no noticeable commonality between the data sets for the crosswind velocities. This is expected, considering the drastic variations of the jet stream and upper atmospheric winds across the United States. As shown in Figure 2, a wide range of velocities are present and vary drastically with altitude. The wind is therefore expected to experience the largest amount of variation within the given data sets. This variation also makes the determination of acceptable crosswind data sets more difficult due to the anticipated fluctuations.

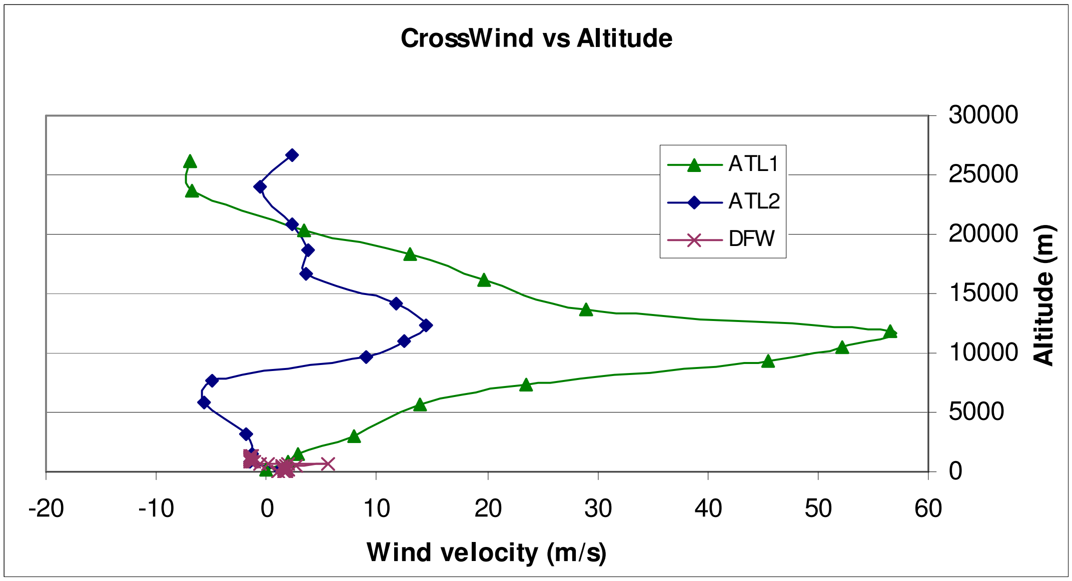


FIGURE 2: VARIATION OF CROSSWIND VELOCITY WITH ALTITUDE

AVAILABLE AIRCRAFT DATA

In order to run the APA Suite to calculate the predicted vortices from each of the meteorological data files provided, the aircraft data (ACDATA) files must be created. The ACDATA files require four numbers:

- Y and Z coordinates (m) of the starting vortex,
- Initial vortex descent rate (m/s),
- Initial vortex separation, b_o (m).

The initial vortex descent rate is defined by $v_o = \dfrac{\Gamma_o}{2\pi b_o}$ (where Γ_o is the initial vortex circulation m^2/s) and is defined as $\Gamma_o = \dfrac{Wg}{\rho U b_o}$ (where W is the aircraft landing weight in kg, g is the gravity constant = 9.81 m/s^2, ρ is air density = 1.2 kg/m^3, and U is the aircraft speed m/s). If the landing weight of an aircraft is unavailable and the initial circulation is unknown, then the weight of the aircraft is assumed to be 90% of the maximum landing weight for each aircraft per Proctor and Hamilton [4].

For MEM, an excel file titled "BY_AIRCR.xls" was included in the data set and documents the aircraft type, altitude, landing speed, weight, and wingspan for each measured vortex case number. This file also provides comments on how the measurement equipment performed during each vortex measurement. For each measured vortex case listed in the BY_AIRCR.xls file, the wingspan (m), altitude (m), weight (kg), groundspeed (m/s), and theoretical calculation of initial vortex strength of the aircraft are extracted to calculate necessary parameters for the ACDATA file. The Y-coordinate of the vortex location is taken as the first Y-coordinate location of the corresponding measured vortex file. The Z-coordinate of the starting vortex is set equal to the altitude taken from the BY_AIRCR.xls file. The initial vortex descent rate is found by dividing the theoretical calculation of initial vortex strength by $2*\pi*b_o$. The wingspan is multiplied by a factor of $\pi/4$ to get the initial vortex separation, b_o. In the event of a missing initial vortex strength value from the BY_AIRCR.xls file, the weight and groundspeed can be used in the initial circulation equation described above to calculate the initial vortex strength. The initial vortex descent rate is then found the same way as described above.

For DFW, the Y-coordinate of the vortex location for each ACDATA file is taken as the first valid Y-position from the corresponding measured vortex file. The Z-coordinate of the starting vortex is assumed to be 100 meters. The initial vortex separation (b_o) is taken from the value calculated for each aircraft located in the "Aircraft Data.xls" spreadsheet. The initial vortex descent rate is calculated by taking the first available vortex value from the measured vortex data files and dividing that value by $2*\pi*b_o$.

The APA Suite requires the value for the initial vortex strength (v_o) to be a positive value. If a negative value is used, the executable assumes an inverted aircraft and will provide incorrect output data values. If any negative value is calculated, the absolute value is taken to ensure a positive number is always used.

AVAILABLE WAKE VORTEX DATA

Wake vortex field measurements were taken at Memphis, TN during the month of December 1994 and August 1995. Wake measurements at Dallas/Fort Worth, TX were taken during the months of November and December 1999. Additionally wake measurements were taken at Idaho Falls in June and September 1990. Each location is discussed separately due to variances in how the files are handled.

MEMPHIS, TENNESSEE

Wake vortex behavior was observed using a mobile continuous-wave (CW) coherent laser Doppler radar (LIDAR) developed at Lincoln Laboratory. For each successful wake vortex measurement, separate starboard and port vortex files are created. The names follow the convention: MEM####P.VOR.ddmmyy_hhmmss, where #### is the case number specific to the Memphis program, and P denotes the port vortex (S for starboard). A note should be made that the day, month, year, hour, minutes, and seconds of the vortex file do not match the actual instance of the vortex measurement time, but instead provide the approximate date and time the file was created by Lincoln Laboratories.

The vortex files are text files where the first data line displays which vortex (port or starboard), airport, site, case number, aircraft type, aircraft model. The second data line contains the LIDAR algorithm version number and the data format version number. The third data line is time of aircraft passage (GMT) as year, month, day, hour, minute, second. Subsequent data lines indicate a vortex location and circulation estimation as seconds relative to aircraft passage, y distance of vortex from LIDAR truck (m), Δy relative y estimation error (m), z altitude of vortex relative to height of LIDAR truck (m), Δz relative z estimation error (m), fd LIDAR focus range – vortex range (m), r_0 cross-range distance between highest positive and negative vortex velocities (m), and $\Gamma(X)$ = average circulation of all measurement points for $X-0.5 < X \leq X+0.5$ m in cross-range distance from the vortex center (m^2/s). Data elements of "9999.0" are invalid [5].

Each vortex file is opened and parsed using a Microsoft Excel Visual Basic script. The script extracts the filename, year, month, day, hour, minute, seconds of analysis time, aircraft type, as well as the Y-position, Z-position and circulation strength at every time step. The circulation strength is calculated as the average circulation from 3-10m radius on each side of the vortex center as recommended by Sarpkaya, Robins, and Delisi [6]. The parsed vortex data, along with the aircraft type, initial time of vortex measurement, and the data flag (described in section 3 below), are saved as an excel spreadsheet named "Results – Measured Vorticies.xls" under the tab "Memphis Measured Vortex Results." Note this data is only the parsed vortex data, not RMS error measures.

DALLAS/FORT WORTH, TEXAS

A wake measurement program was performed at the Dallas-Fort Worth International Airport in November and December of 1999. The wake vortex measurements were collected through the use of three different sensors operated by three different entities. MIT Lincoln Laboratory performed the design, development, and deployment of the wake vortex tracking continuous wave (CW) LIDAR. The Volpe Transportation Systems Center provided a ground windline and NASA Langley operated a pulsed LIDAR for wake vortex tracking.

The measured vortex data files collected at DFW follow a similar naming format: "sensorname_owner.yymmdd_hhmmss" where sensor name is "lidar" for files produced by lidar and "wline" for files produced by the wind line. Owner is "N" for NASA, "L" for Lincoln Labs, and "VN" for Volpe (e.g., wline_VN.991110_000159). Each file is a text file beginning with comment lines (denoted by "#"). These comment lines provide information pertaining to type of aircraft, time of vortex measurement, number of lines of data recorded, and time as well as Y- and Z-coordinates and an average of the circulation strength of a 3-10m radius of both the port and starboard vortices. Data elements of "9999" are invalid [7].

As discussed earlier, each vortex file is opened and parsed using a Microsoft Excel Visual Basic script that extracts the filename, year, month, day, hour, minute, seconds of analysis time, and the aircraft type. The Y-position, Z-position and circulation strength of both the starboard and port vortex are also extracted at each time step. The parsed vortex data, along with the aircraft type, initial time of vortex measurement, and the data flag (described in section 3 below), are saved as an excel spreadsheet named "Results – Measured Vorticies.xls" under the tab "Dallas Measured Vortex Results." Note this data is only the parsed vortex data, not RMS error measures.

IDAHO FALLS, IDAHO

Meteorological data was collected at altitudes between 0-12,500m at the airport in Idaho Falls, Idaho during the months of June and September in 1990. The instruments used to collect meteorological and wake data included LDV-Lidar, a Monostatic Acoustic Velocimeter System, and sounding equipment [4]. The measured meteorological files follow a naming format of the form prof_lvl1_IDF.##_yyyymmdd_hhmmss where ## is 00, 09, or 12 depending on which push the data was recorded, yyyymmdd is the date, and hhmmss is the time. Vertical profiles for both the u and v wind components, temperature, and pressure are included each data file. Only 4 cases of wake data recorded at IDF were provided for analysis. These cases range between 0 and 150m altitude. Upon further analysis of the data and consultation with George Green and Jim Hallock at VOLPE the data was considered to be too controversial for use in this analysis and was discarded.

VORTEX DATA QUALITY CONSIDERATIONS

If at any time step an invalid data point "9999.0" or an empty data point is encountered, the data for that time step is discarded. To facilitate a comparison of this measured data to the APA Suite predicted vortex measurements, the measured data is categorized by quality. The first time step of the measured file is always retained regardless of the fidelity of the first time step. If the first time step contains all valid data points, then the file is flagged with a "D" denoting the data coincides with the first time recorded by the measurement systems. If the initial time steps are discarded due to invalid data points but subsequent time steps are valid, then the file is flagged with a "T" denoting the data is truncated with respect to the overall measurement time. If the measured vortex data file contains no valid data points or only one valid data point, then the file is flagged with an "E" denoting the data set is empty and unusable.

METHOD

The sensitivity study undertaken here had several steps:

1. Convert the available meteorological data into files following APA Suite specifications
2. Convert each meteorological data file into resolutions of interest (1m–100m)
3. Run each set of meteorological data files using the APA Suite (both TDP and APA engines)
4. Compare the APA Suite predicted files to the measured wake vortex files to determine the accuracy of the APA Suite Vortex Predictions
5. Compare the APA Suite predicted files to determine the sensitivity of the APA Suite vortex predictions to the input file resolution fidelity

A MATLAB script has been written under Task 1 to automate the process of parsing different types of meteorological data files into a format easily read into the APA Suite software. This script also stores the output in a form for routine automated post-processing. A general overview of the MATLAB script is shown in Figure 3.

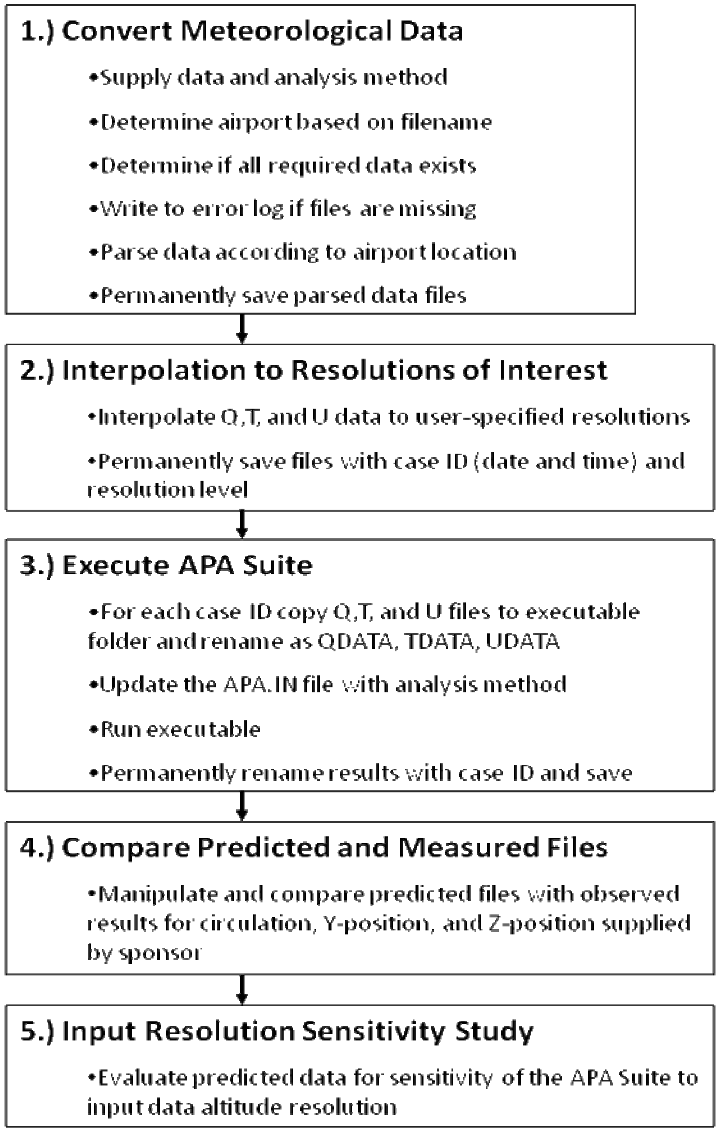


FIGURE 3: OVERVIEW OF THE AUTOMATED PROCESS FOR ANALYSIS

METEOROLOGICAL DATA INPUT AT RESOLUTIONS OF INTEREST

An objective of this research under Task 1.1, is to determine the vertical data resolution required for reasonable vortex prediction. Unfortunately, the raw data files available did not always have appropriate or even consistent levels of resolution. It was necessary for the raw data files to be interpolated or made sparse as necessary to achieve a consistent resolution. In the present study, the data files are interpolated at user-specified resolutions of 1, 5, 15, 20, 25, 30, 40, and 100m using MATLAB's `interpolation` function. These files follow the same naming convention as the original parsed meteorological data files with the interpolation value added to the end (e.g., TDATA_YYMMDD_HHMMSS_MEM20).

Input files are saved according to the convention ACDATA_YYMMDD_HHMMSS_DFW#GT#### where the ACDATA tells the file type, the first six numbers are the date, the second six numbers are the time, the three letter designation represents the airport (DFW, MEM) followed by the interpolated resolution, and the GT#### represents the corresponding vortex file identifier. Each vortex file is labeled with a unique GTID. Meteorological data files that are measured at the same time or very close to the time of the vortex measurement are assigned the same GTID as the corresponding vortex file. This enables a faster and more intuitive process to match a specific meteorological data file with the appropriate vortex data file.

In addition to the input data from DFW and MEM additional data was collected for Atlanta at altitudes 0-20,000m (above 1,400m available from DFW and MEM). From available meteorological data from Atlanta, the UDATA, TDATA, and QDATA files are created. The QDATA files are created with a constant profile of $1x10^{-4}$ m^2/s^3 with an altitude equal to that of the highest altitude for the temperature and crosswind profile. For each of the U, T, and Q Data files, the values are interpolated into the following resolutions: 1, 5, 15, 20, 25, 30, 40, and 100m. Each of these resolutions is evaluated by the APA Suite for future use.

COMPARISON OF PREDICTED AND MEASURED DATA

Once the data for the predicted and measured vortices are parsed and catalogued two comparisons were performed. The first compares the measured wake vortex data to the APA Suite predictions. Matlab scripts have been written to combine the two sets of data and perform a Root Mean Square (RMS) analysis of the data. The first script, titled “rmscalc.m”, stores all the measured and predicted vortex time series data into mat-files. It subsequently calls a script called getrms.m, which computes the normalized mean RMS of the vortex strength, Y-position, and Z-position. This approach is similar to what was done by Sarpkaya, Robins, and Delisi [6].

This script requires inputs of: a database Excel file containing a list of measured vortex data and the corresponding predicted vortex files “Aircraft Data.xls” an Excel file containing the measured vortex data “Results-Measured Vortices.xls”, and an Excel file containing the predicted vortex data “Results-Predicted Vortices_DFW_Only.xls” and “Results-Predicted Vortices_MEM_Only.xls”. The output of this script are a .mat file containing the data in the measured and predicted vortex Excel files and a .mat file containing the resulting mean RMS values for each case.

The “getrms.m” script is used by the previously discussed “rmscalc.m” script. This function accepts the measured and predicted time series data for a particular case, along with the initial circulation, as input. The input data is first checked to see if there is any anomalous data, such as a repeated measurement time or a dead vortex. Any measurement data that is repeated is removed. A dead vortex is defined as a case where the predicted circulation values do not go down to zero, but instead settle to a constant value. If a dead vortex is encountered, all data that remains constant is removed for the RMS calculation, and a dead vortex flag is returned along with the results. A dead vortex flag is represented by a 1 under the column “Dead Vortex?” within the Results spreadsheet. If the measured data does not begin at the same time as the predicted data, it is shifted for correction. The measured data is shifted to the first available time step provided within the measured vortex files. This time step is listed for every file within the “Results-Measured Vortices.xls” file along with the data flags (T, D, and E) described earlier in this report. If a data flag of “E” is encountered for any vortex case, that case is discarded. Once the measured data is shifted to the corrected starting time, both the measured and predicted data are linearly interpolated to the same time vector for comparison. The normalized RMS value is calculated for the circulation, Y-position, and Z-position of each vortex case as follows:

$$\overline{x}_{MSE}^{norm} = \frac{1}{\left|x_{m0}\right|}\sum_{i=1}^{n}\frac{\left(x_{mi}-x_{pi}\right)^2}{n} \tag{1}$$

Here, n is the number of data points in the linearly interpolated data set, x_{mi} is the measured data at a particular time, x_{pi} is the predicted data at the same time, and x_{m0} is the initial measured data. The variable x can represent circulation, Y-position, or Z-position. This calculation is performed for every time step existing for both the measured and predicted vortices. These RMS values are stored as a .mat file.

A script called "rms2xls.m" reads the RMS .mat files and writes them to a .xls file. For completeness, and to aid in Task 2.2, a script called "writeall.m" is created to combine the RMS.mat files, the measured vortex data .mat files, and the predicted vortex data .mat files into one single spreadsheet called "Results_All_20100507.xls". This spreadsheet documents every predicted vortex data file name, the corresponding measured vortex file name, type of analysis (APA/TDP), GTID, airport location, aircraft type, aircraft weight class (large, small, etc.), aircraft weight, aircraft wingspan, initial circulation, weather condition, wind and cloud quantifiers, the measured vortex data, predicted vortex data for all interpolated resolutions, and the corresponding MSEnorm values of circulation, Y-position, and Z-position for each of the predicted vortex interpolated resolution. The spreadsheet is divided by airport (DFW/MEM) and analysis type (APA/TDP) to facilitate fitting all the data within a single Excel file.

COMPARISON OF PREDICTED DATA AT DIFFERENT INPUT RESOLUTIONS

The second comparison performed an analysis of the effect of input resolution on the APA Suite predictions. The RMS of the predicted vortex strength, Y-position, and Z-position for height resolutions of 1, 5, 15, 20, 25, 30, 40, and 100m were calculated, then normalized by the values for the 1m resolution. This process is described by the following equations, where RMS^{norm} represents the RMS of the particular variable, normalized by that for the 1m spacing:

$$RMS_{circ}^{norm} = \frac{\sqrt{\dfrac{\sum_{i=1}^{n} circ_i^2}{n}}}{\sqrt{\dfrac{\sum_{i=1}^{n} circ_{i,1m}^2}{n}}},\ RMS_{Y}^{norm} = \frac{\sqrt{\dfrac{\sum_{i=1}^{n} Y_i^2}{n}}}{\sqrt{\dfrac{\sum_{i=1}^{n} Y_{i,1m}^2}{n}}},\ RMS_{Z}^{norm} = \frac{\sqrt{\dfrac{\sum_{i=1}^{n} Z_i^2}{n}}}{\sqrt{\dfrac{\sum_{i=1}^{n} Z_{i,1m}^2}{n}}} \tag{2}$$

Here, n is the number of linearly interpolated data points. Using the resulting values, the degree to which resolution degrades prediction precision can be ascertained. The values from 0.95 to 1.05

constitute those that lie within 95% of the values for the 1m spacing. Thus, a determination can be made as to at what spacing the RMS exceeds this allowable error.

RESULTS

COMPARISON OF PREDICTED AND MEASURED DATA

DATA CHARACTERIZATION

In addition to the categorization of results by MSE^{norm}, other statistical analyses may be done to characterize the distributions of the dependent variables. Histograms and box plots have been made to evaluate the data. Histograms are useful in evaluating the distribution of data for each dependent variable. Each histogram plotted has been created to show the same number of standard deviations.

Box plots are another useful way to visually evaluate the data. The central mark of each box is the median and the box edges represent the 25th and 75th percentiles. Whiskers extend to the most extreme data points not considered outliers, and outliers are plotted individually. All of the box plots here are notched where the notch represents a confidence interval. When comparing two box plots, the medians of the data sets are different with a 95% confidence interval (CI) if the notches do not overlap. In some cases where the sample size is small, the notched intervals extend beyond the edge of the box.

In the results file there were 2,634 cases. These results were filtered to remove cases where the data flag was not equal to D and cases where vortex death occurred. Cases that exhibited data flags of T (truncate and displace) or E (single data point empty) were eliminated. Cases where vortex death occurred at 1m resolution were also eliminated. Finally cases where vortex death occurred at 100m resolution were eliminated. After removing these cases, the remaining 1,129 data sets were analyzed. Two different data sets were created and evaluated. The first data set includes all points where the data flag is D (displace only) and vortex death did not occur. The second data set is a more limited set where outliers from the first set were eliminated. Outliers are defined here as points that lie outside 1.5σ or 1.5 standard deviations. For each dependent variable data set (circulation, Y-position, and Z-position) this criterion of 1.5σ was used to eliminate the outliers.

Figure 4 shows a comparison of the two data sets for 1m resolution. The graph on the left contains histograms for the dependent variables with the outliers included while the graph on the right contains the histograms with the outliers excluded. Removing the outliers was a necessary step in to allow parametric analysis and additionally, allows the variations in the normal range of data to be better observed. Comparing the number of points in each bin and the distribution of bins between the left and right plots for each dependent variable, one can see that the Y-position distribution is the most strongly affected by outliers. Figure 5 shows box plots for the dependent variables with the outliers included (left) and excluded (right). Again, once the outliers were removed, more information was observed from the results such as the median and 25th and 75th percentiles.

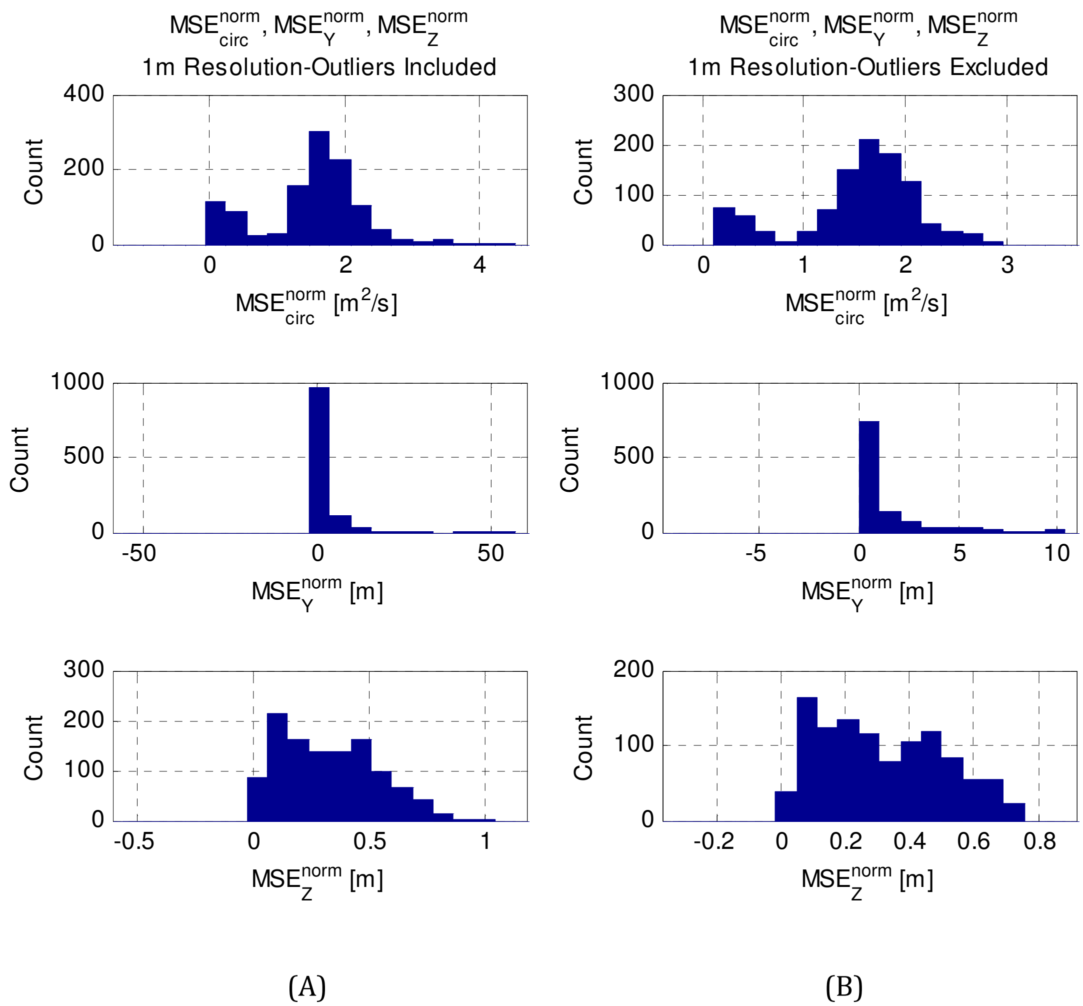


FIGURE 4. HISTOGRAMS FOR 1M RESOLUTION. (A) OUTLIERS INCLUDED (B) OUTLIERS EXCLUDED

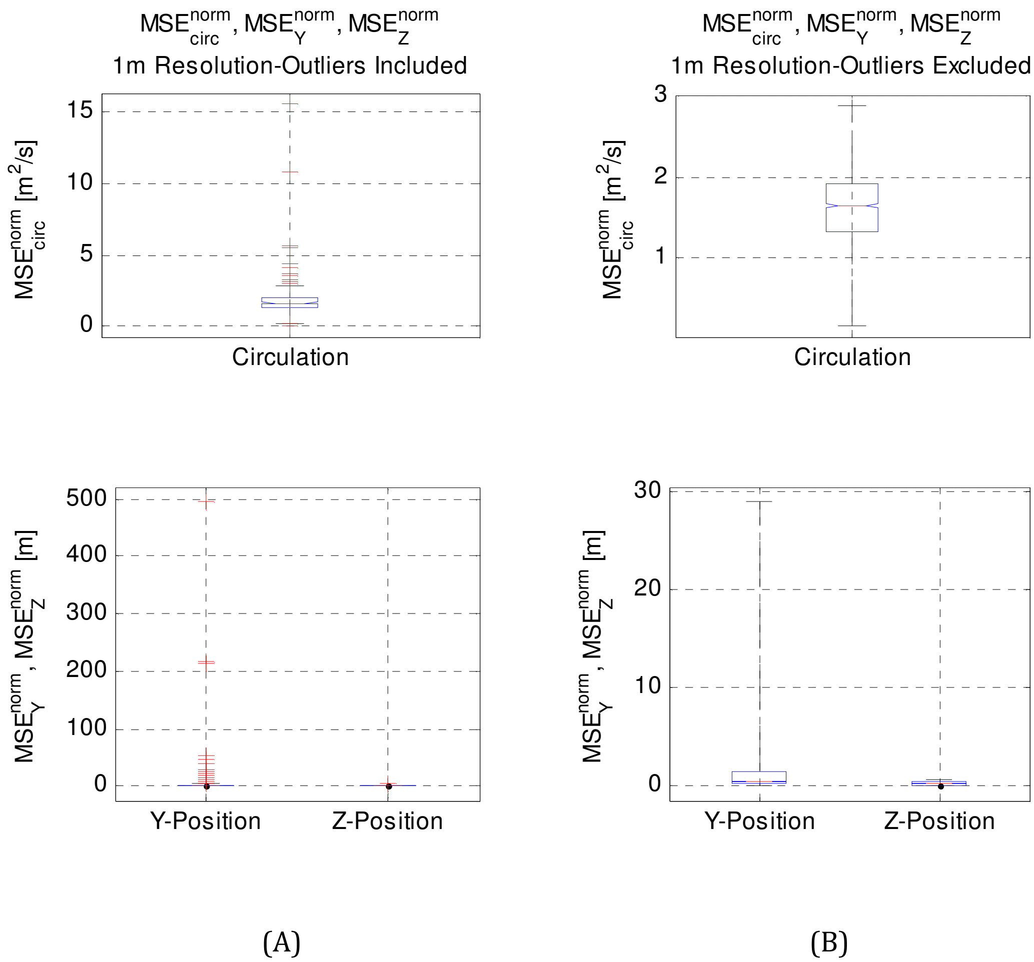
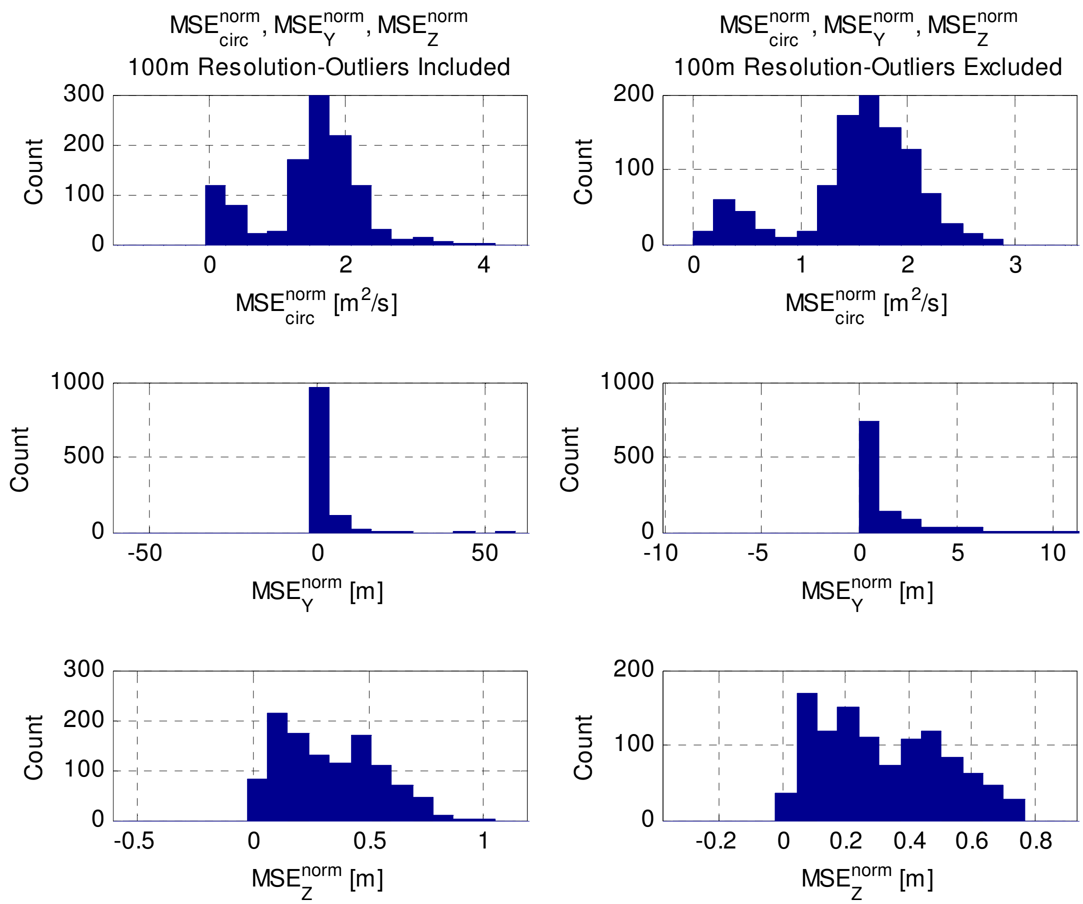


FIGURE 5. BOX PLOTS FOR 1M RESOLUTION. (A) OUTLIERS INCLUDED (B) OUTLIERS EXCLUDED

Figure 6 shows the histograms for 100m resolution. There is not a significant difference observed between the 1m resolution and the 100m resolution plots. The overall distribution for each dependent variable is the same. However, removing the outliers does affect the number of points in each bin as well as the distribution of bins. Figure 7 shows the box plots for 100m resolution with outliers included (left) and outliers excluded (right).



(A) (B)

FIGURE 6. HISTOGRAMS FOR 100M RESOLUTION. (A) OUTLIERS INCLUDED (B) OUTLIERS EXCLUDED

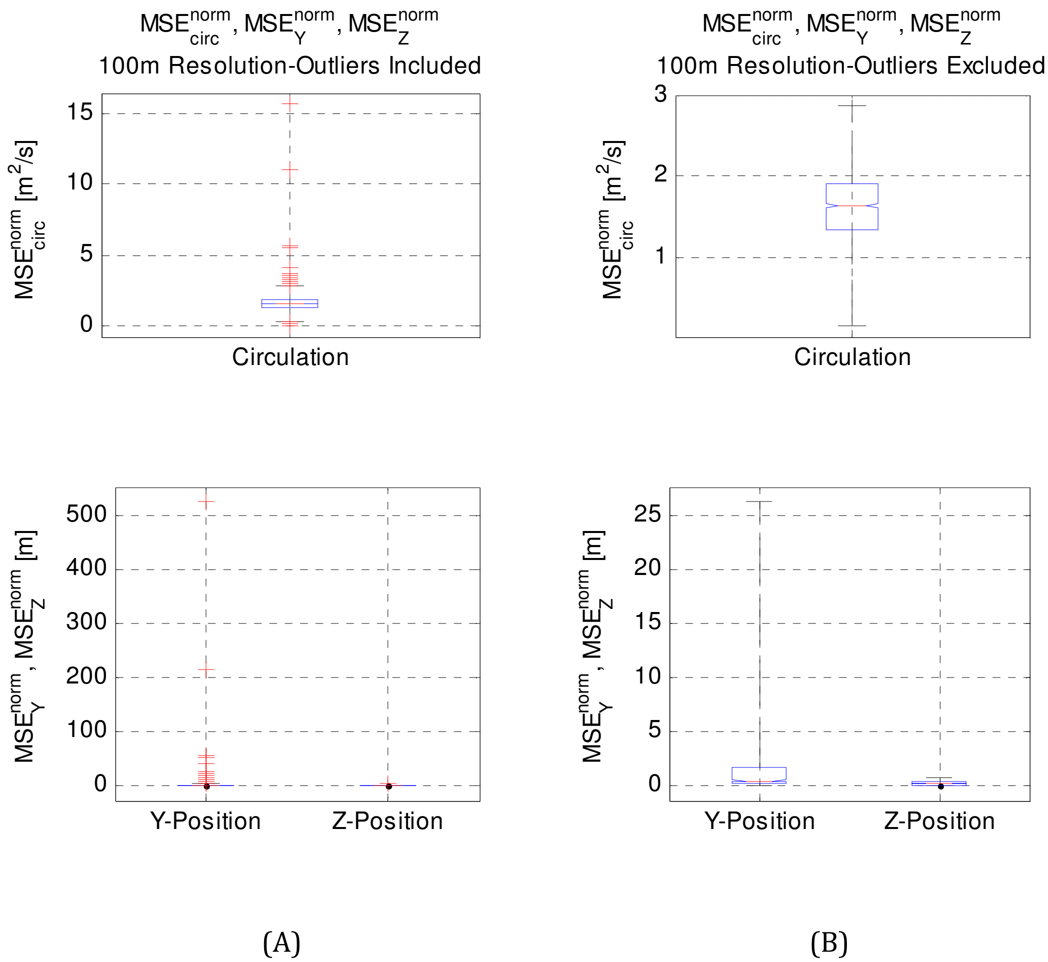


FIGURE 7. BOX PLOTS FOR 100M RESOLUTION. (A) OUTLIERS INCLUDED (B) OUTLIERS EXCLUDED

Figure 8 shows a side-by-side comparison of the outlier-excluded cases for 1m and 100m resolutions. Circulation shows the largest differences while Z-position shows almost no change. Near zero, the circulation histogram shows higher counts at 100m resolution than at 1m resolution. Also, between 1.5 m^2/s and 2 m^2/s the circulation counts appear more constant for the 100m case. Figure 9 shows box plots for comparing the 1m and 100m cases with outliers excluded. The only noticeable difference observed between the cases is the value of the upper whisker on the Y-position, which is slightly larger for the 1m case.

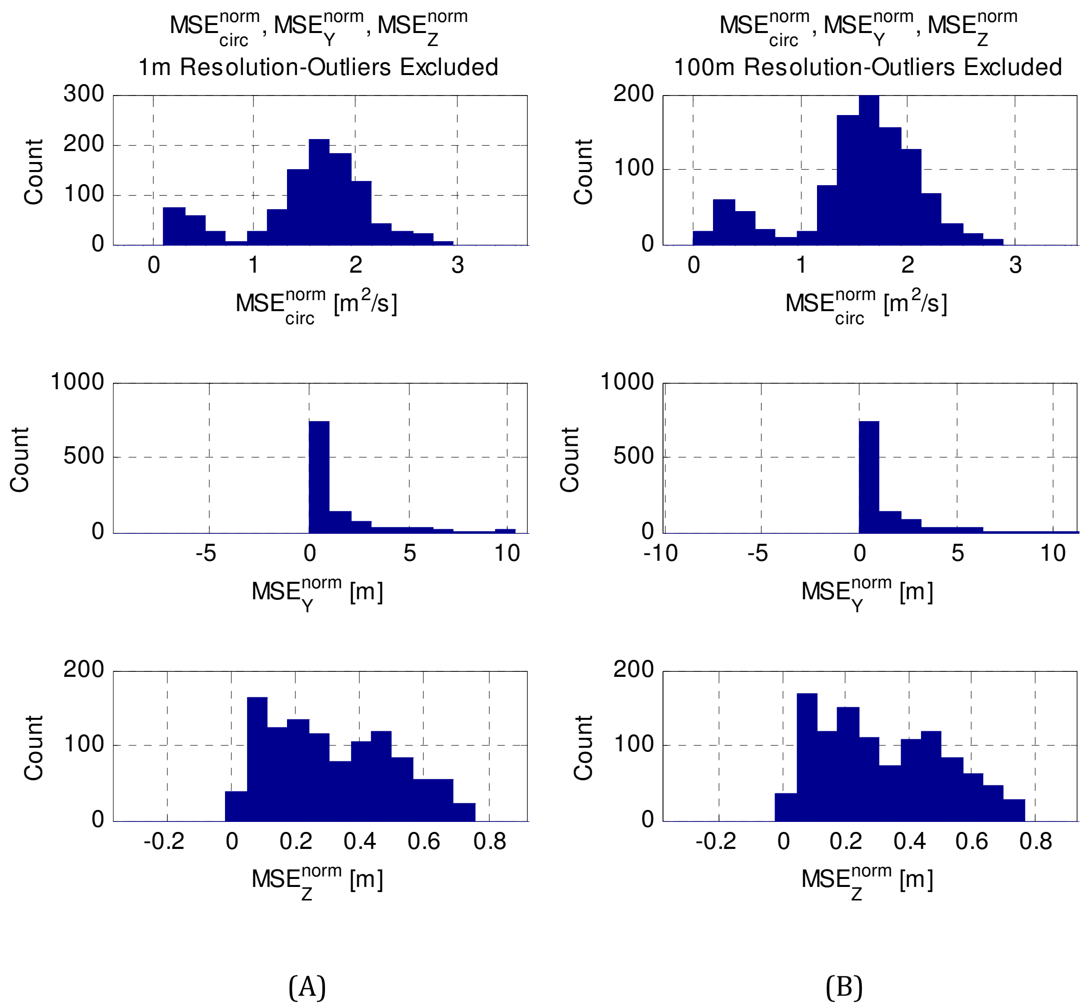
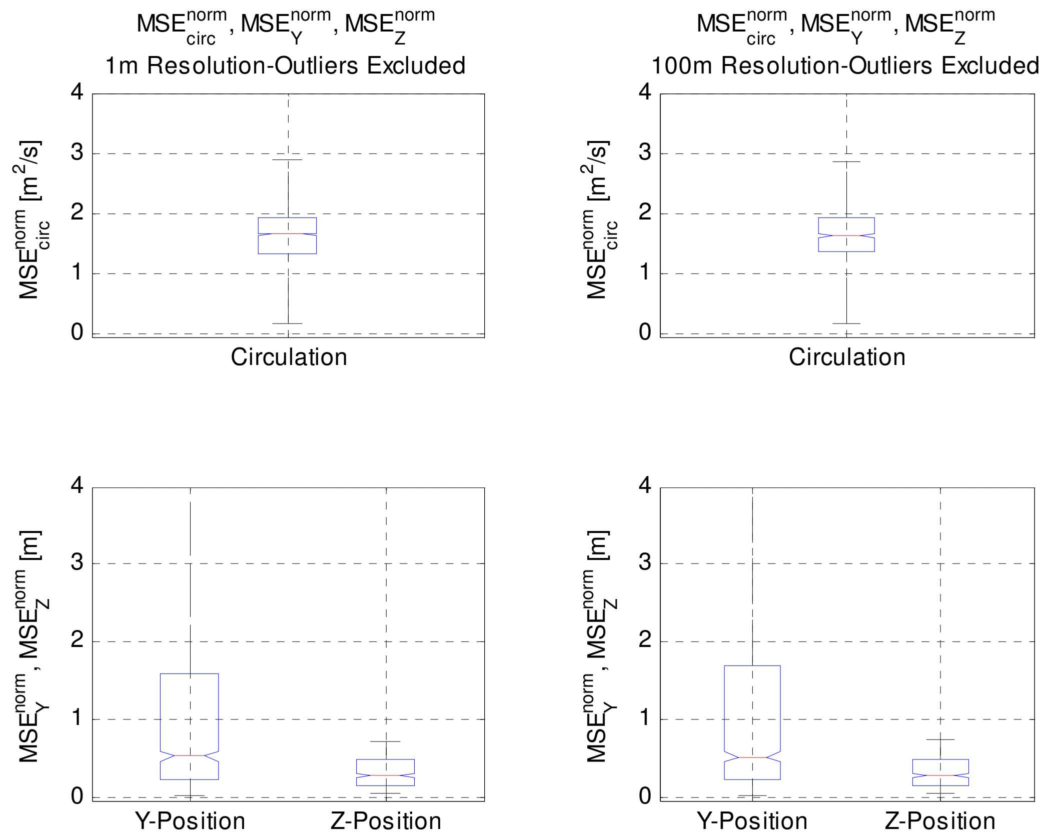


FIGURE 8. HISTOGRAMS WITH OUTLIERS EXCLUDED. (A) 1M RESOLUTION (B) 100M RESOLUTION



(A) (B)

FIGURE 9. BOX PLOTS WITH OUTLIERS EXCLUDED. (A) 1M RESOLUTION (B) 100M RESOLUTION

Based on the results of the data characterization studies detailed above, no significant differences were found between the 1m and 100m resolution cases. As a result of this conclusion, further comparisons of the measured and predicted data presented in this report are only shown for 1m resolution with outliers excluded.

The next section of the report describes the overall observed accuracy of the APA Suite after completing data analysis on the measured and predicted results. Results for each dependent variable are analyzed to evaluate the influences of the independent variables and discussed in subsequent sections. Independent variables are analysis model, location, aircraft weight class, winds, and cloud cover. Combinations of independent variables including: analysis model and winds, analysis model and cloud cover, and winds and cloud cover are also investigated to determine any impact on the dependent variables.

OVERALL ACCURACY OF APA SUITE

The overall accuracy of the APA Suite can be evaluated by looking at the effects of the different independent variables on the dependent variables of circulation, Y-position, and Z-position. Each dependent variable is considered and presented separately in the following sections. However, a summary of the conclusions is presented in this section.

The independent variables are analysis model, location, aircraft weight classification, winds, and cloud cover. Combinations of analysis model and winds, analysis model and cloud cover, and winds and cloud cover are also considered. Surprisingly, analysis model is an independent variable that was not observed to significantly affect the APA Suite prediction of the dependent variables based on which model was used. Location (MEM or DFW) was found to affect the predictions of each dependent variable. Winds, cloud cover, and the combination of winds and cloud cover were found to affect the prediction of Y- and Z-position but not circulation. Additionally, aircraft weight class was only found to affect the Y-position predictions. The weather conditions affecting the position predictions of the APA Suite was expected since any environmental changes such as crosswind shear during measurement affect the motion of the vortex.

Table 2 shows a summary of the dependent variables and provides the minimum and maximum as well as the 1st and 3rd quartiles and the median for each data set. The values listed in the table are RMS values between the measured and predicted data.

A paired T-test was used to compare the normalized MSE data at both resolutions (Table 3). The results of this test show that the circulation and Z-position are significantly different between the two resolutions, with p-values of 0.0005 and 0, respectively, but the Y-position difference is not significant. Based on the results of this T-test, further comparisons of the measured and predicted data presented in this report are only shown for both resolutions with outliers excluded.

TABLE 1. SUMMARY OF THE OVERALL APA SUITE ACCURACY WITH ROOT MEAN SQUARE ERROR (OUTLIERS INCLUDED).

Parameter	1m Resolution			100m Resolution		
	Circulation RMSE (m^2/s)	Y-Position RMSE (m)	Z-Position RMSE (m)	Circulation RMSE (m^2/s)	Y-Position RMSE (m)	Z-Position RMSE (m)
Mean	18.1411	7.3473	5.4204	18.1032	8.9303	5.4337
Std. Deviation	13.3270	22.8710	4.4893	13.2923	33.5420	4.5280
Minimum	1.5463	0.7325	1.1868	0.6872	1.4309	1.0672
Maximum	32.6343	134.5068	10.1072	33.0885	210.5918	10.1095

TABLE 2. SUMMARY OF DEPENDENT VARIABLES (OUTLIERS EXCLUDED)

Parameter	1m Resolution			100m Resolution		
	Circulation MSE^{norm} (m^2/s)	Y-Position MSE^{norm} (m)	Z-Position MSE^{norm} (m)	Circulation MSE^{norm} (m^2/s)	Y-Position MSE^{norm} (m)	Z-Position MSE^{norm} (m)
Mean	1.5335	1.7039	0.3160	1.5480	1.7309	0.3196
Std. Deviation	0.6101	3.0965	0.1919	0.5770	3.1801	0.1954
Minimum	0.1596	0.0148	0.0282	0.1719	0.0146	0.0275
1st Quartile	1.3316	0.2203	0.1361	1.3519	0.2184	0.1354
Median	1.6439	0.5245	0.2742	1.6356	0.5155	0.2746
3rd Quartile	1.9144	1.5965	0.4729	1.9179	1.7013	0.4810
Maximum	2.8821	29.0584	0.7234	2.8707	26.3737	0.7366

TABLE 3. RESULTS OF T-TEST BETWEEN 1M AND 100M MSE^{NORM} VALUES (OUTLIERS EXCLUDED)

Parameter	Circulation	Y-Position	Z-Position
p-value	0.0005	0.8021	0
t-statistic	3.5096	-0.2507	-6.2847
Degrees of Freedom, d.f.	1014	1121	1101
Estimated population std. deviation of difference (1m – 100m)	0.0695	1.1191	0.0155

INFLUENCES ON THE CIRCULATION ACCURACY OF APA SUITE

Table 4 is a summary of an Analysis of Variance (ANOVA) table. An ANOVA is a method for testing the influences of multiple factors on the mean of the dependent variable, which is circulation in this case. The first five rows of

Table 4 are to evaluate the influence of each independent variable on the circulation and the last three rows evaluate the influence of interactions between independent variables on the circulation. The first column represents the source of the variability, the second column represents the F statistics which are the ratios of the mean squares, and the third column represents the p-value for the F statistics. The second and third columns are shown for 1m and 100m resolutions.

Further analysis from the results of the ANOVA is conducted for influences that are deemed to be significant. If any p-value is near zero, the influence factor could be statistically significant. For the purposes of this report, any p-value less than 0.01 is considered statistically significant and the influence of the corresponding independent variable is investigated. As

Table 4 shows, the only influence that is statistically significant for circulation is the location.

TABLE 4. INFLUENCES OF INDEPENDENT VARIABLES ON CIRCULATION

Influences on Circulation	**d.f.**	**1m Resolution**			**100m Resolution**		
		r^2	**F**	**Prob > F**	r^2	**F**	**Prob > F**
Analysis Model (APA or TDP)	1	0.0009	1.15	0.2833	0.0015	1.83	0.1768
Source (DFW or MEM)	1	0.2363	288.06	0	0.2329	278.69	0
Aircraft Weight Class (Heavy, Large, or Small)	2	0.0005	0.31	0.7361	0.0015	0.89	0.4111
Winds (Windy or Calm)	1	0	0.04	0.8426	0.0003	0.33	0.5667
Cloud Cover (Cloudy or Clear)	1	0.0005	0.58	0.4483	0.0001	0.14	0.7078
Analysis Model and Winds	1	0.0005	0.65	0.4199	0.0001	0.08	0.7714
Analysis Model and Cloud Cover	1	0	0	0.9723	0	0.01	0.9826
Winds and Cloud Cover	1	0.0015	1.83	0.1762	0.0006	0.68	0.4094
Error	913						

Figure 10 shows box plots for RMS circulation data from DFW and MEM, respectively. The medians of the data sets are different by a 95% confidence interval since the notches do not overlap. The MEM data is also observed to cover over a smaller range than the DFW data, since the 25th and 75th percentiles for MEM are closer together than for DFW. Thus, DFW data has a larger variance.

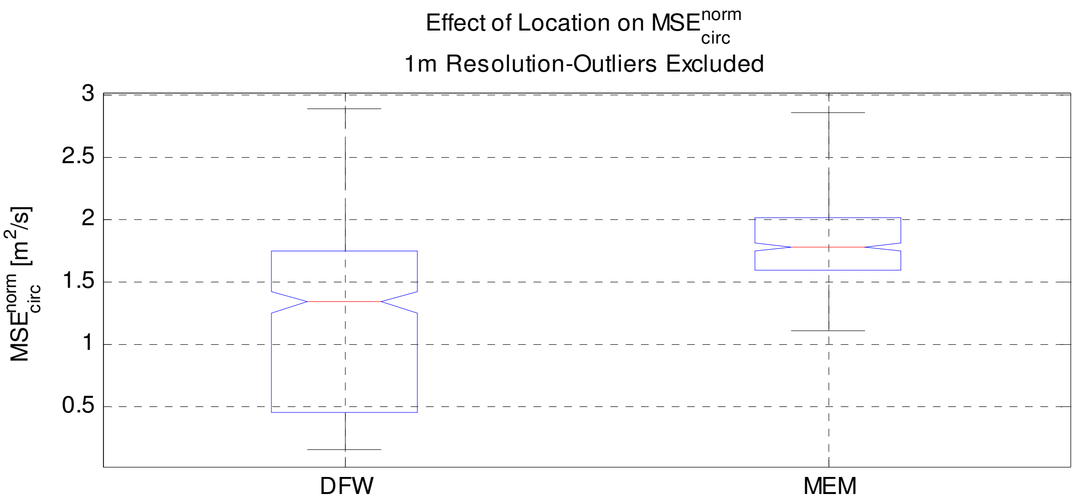


FIGURE 10. INFLUENCES ON CIRCULATION

Planned contrasts are completed for each dependent variable using Matlab's `multcompare` command and the results from the corresponding ANOVA. The output of the planned contrasts can be displayed as shown in Table 5 for circulation. The groups indicate the data sets to compare, the 95% confidence interval (CI) for the true mean represents a range for the true difference between the means, and the significance indicates if zero is contained in the confidence interval. If zero is not contained in the confidence interval, the difference between the means is significant at the 0.05 level. If the confidence interval does contain zero, the difference between the means would not be significant at the 0.05 level. Table 5 shows the planned contrast results for circulation at 1m resolution and shows that only location resulted in a confidence interval that did not contain zero for circulation.

TABLE 5. PLANNED CONTRAST RESULTS FOR CIRCULATION AT 1M RESOLUTION

Group 1	**Group 2**	**95% CI for True Mean**	**Significant?**
APA	TDP	[-0.1998 0.0584]	No
DFW	MEM	[-0.6793 -0.5387]	Yes
Calm	Windy	[-0.1655 0.2028]	No
Clear	Cloudy	[-0.1148 0.2597]	No

INFLUENCES ON THE Y-POSITION ACCURACY OF APA SUITE

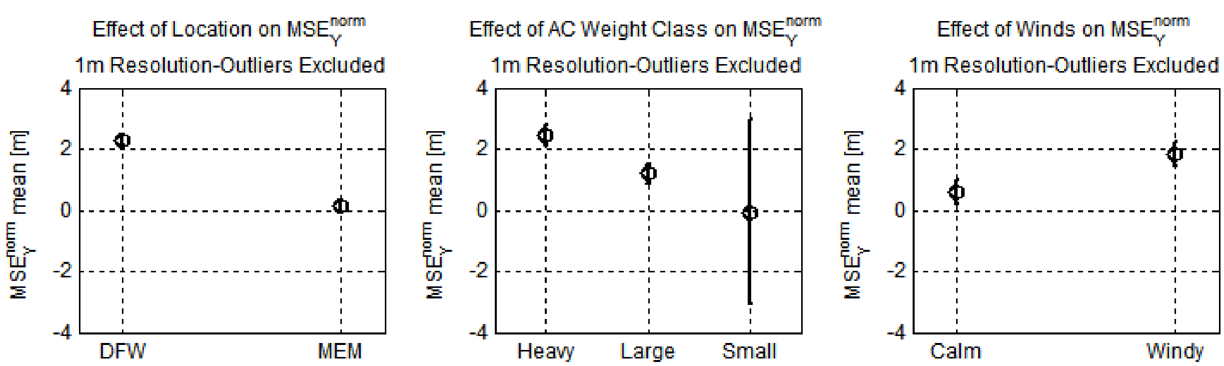
The second dependent variable considered is Y-position. Table 6 shows the ANOVA results and indicates that location, aircraft weight class, winds, cloud cover at 1m resolution, and the combination of winds and cloud cover significantly influence the Y-position accuracy of the APA

Suite. Interestingly, cloud cover only plays a significant role (p-value < 0.01) for 1m resolution and not 100m resolution, although the p-value for 100m resolution is only 0.0513.

TABLE 6. INFLUENCES OF INDEPENDENT VARIABLES ON Y-POSITION

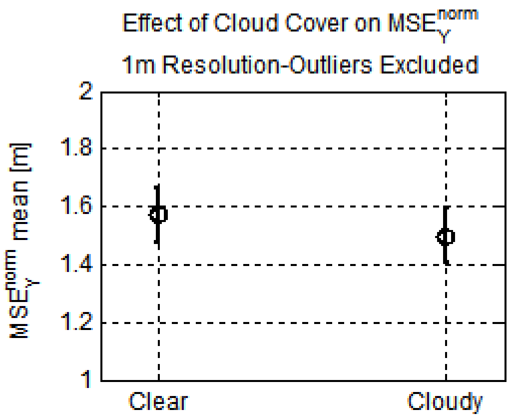
Influences on Y-Position	**d.f.**	**1m Resolution**			**100m Resolution**		
		r^2	**F**	**Prob > F**	**r^2**	**F**	**Prob > F**
Analysis Model (APA or TDP)	1	0	0	0.949	0.0001	0.16	0.6933
Source (DFW or MEM)	1	0.1148	133.32	0	0.0924	103.62	0
Aircraft Weight Class (Heavy, Large, or Small)	2	0.0203	11.77	0	0.0113	6.32	0.0019
Winds (Windy or Calm)	1	0.0079	9.23	0.0024	0.0067	7.5	0.0063
Cloud Cover (Cloudy or Clear)	1	0.0036	4.23	0.04	0.0034	3.81	0.0513
Analysis Model and Winds	1	0	0	1	0.0001	0.09	0.7614
Analysis Model and Cloud Cover	1	0.0003	0.26	0.6075	0.0002	0.26	0.6132
Winds and Cloud Cover	1	0.0065	7.5	0.0063	0.0063	7.11	0.0078
Error	995						

Figure 11 shows the effects of location, aircraft weight class, and winds on the MSE^{norm} Y-position. Again, MEM demonstrates much smaller variance. Although difficult to see in the figure, the notches do not overlap. For aircraft weight class, the greatest variance is seen for "heavy" classification, and the least variance is seen for "small" designation. Figure 11 (C) shows that calm conditions versus windy conditions impact the Y-position with larger variance. Figure 12 shows the effects of cloud cover and the combination of winds and cloud cover. Clear conditions versus cloudy conditions are observed to impact the Y-position in Figure 12 (A). Based on this information, the combination of calm/clear conditions is expected to have the most significant impact and the largest variance. Figure 12 (B) confirms this expectation and reveals that calm/cloudy conditions have the smallest variance.

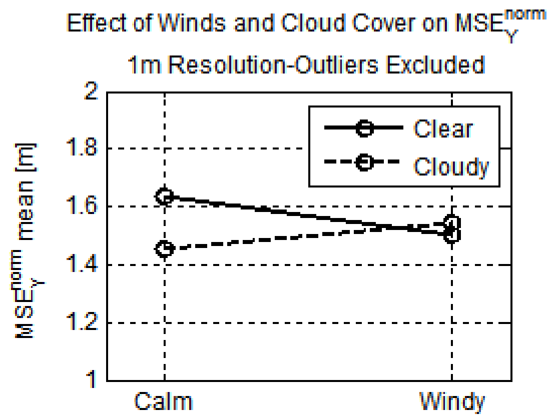


(A) (B) (C)

FIGURE 11. INFLUENCES ON Y-POSITION. ERROR BARS SHOW 99% CI. (A) LOCATION (B) AIRCRAFT WEIGHT CLASS (C) WINDS



(A)



(B)

FIGURE 12. INFLUENCES ON Y-POSITION. ERROR BARS SHOW 99% CI. (A) CLOUD COVER (B) WINDS AND CLOUD COVER

Table 7 shows the planned contrast results for Y-position at 1m resolution and confirms that location, wind conditions, and cloud cover have significant differences at the 0.05 level. The analysis model is not found to be significant because zero is contained on the confidence interval.

TABLE 7. PLANNED CONTRAST RESULTS FOR Y-POSITION AT 1M RESOLUTION

Group 1	Group 2	95% CI for True Mean	Significant?
APA	TDP	[-0.6754 0.7210]	No
DFW	MEM	[1.8099 2.5499]	Yes
Calm	Windy	[-2.0215 -0.4361]	Yes
Clear	Cloudy	[0.0396 1.6497]	Yes

INFLUENCES ON THE Z-POSITION ACCURACY OF APA SUITE

The final dependent variable considered is Z-position and the summary of the ANOVA is shown in ent variables are investigated.

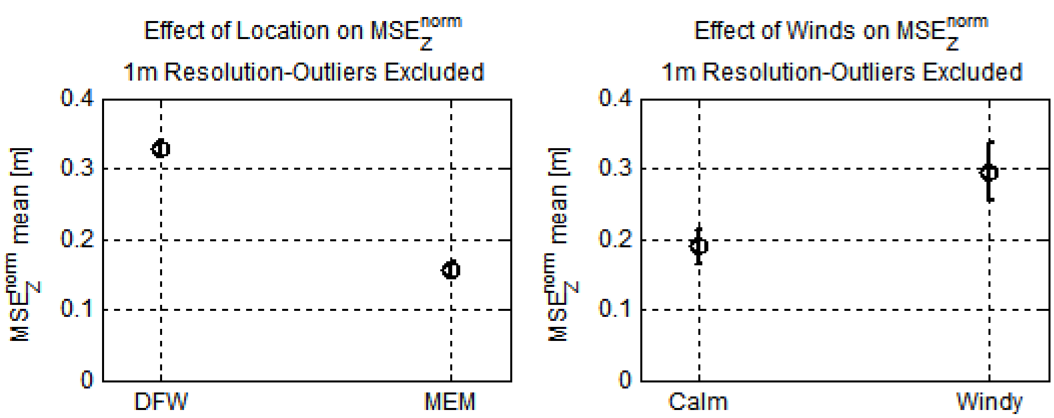
Table 8. Again, location is determined to have a significant impact on the results. Additionally, winds, cloud cover, and the combination of winds and cloud cover impact the results. Each of these independent variables are investigated.

TABLE 8. INFLUENCES OF INDEPENDENT VARIABLES ON Z-POSITION

Influences on Z-Position	d.f.	1m Resolution			100m Resolution		
		R^2	F	Prob > F	R^2	F	Prob > F
Analysis Model (APA or TDP)	1	0.0009	1.19	0.2757	0.0008	1.13	0.2884
Source (DFW or MEM)	1	0.1838	241.19	0	0.1938	258.57	0
Aircraft Weight Class (Heavy, Large, or Small)	2	0.0018	1.17	0.3123	0.0013	0.86	0.422

Winds (Windy or Calm)	1	0.0155	20.39	0	0.0144	19.16	0
Cloud Cover (Cloudy or Clear)	1	0.0389	51.05	0	0.0369	49.29	0
Analysis Model and Winds	1	0.0005	0.7	0.4022	0.0002	0.32	0.5723
Analysis Model and Cloud Cover	1	0.0016	2.16	0.1424	0.0011	1.52	0.2173
Winds and Cloud Cover	1	0.0210	27.58	0	0.0215	28.74	0
Error	981						

Figure 13 shows the effects of location, aircraft weight class, and winds on Z-position. According to Figure 13 (A), DFW and MEM have the same overall range for RMS Z-position values but the medians are statistically different because the 99% CI ranges do not overlap. Figure 13 (B) shows that calm and windy conditions both influence Z-position with similar ranges of variance. Figure 14 shows the effects of cloud cover and the combination of winds and cloud cover. Clear conditions influence Z-position slightly more than cloudy conditions based the box plot variances in Figure 14 (A). Calm/clear conditions or windy/clear conditions are expected to influence Z-position. Figure 14 (B) confirms that calm/clear conditions and windy/clear conditions demonstrate the largest ranges in results. However the other combinations of conditions also have large variances.



(A) (B)

FIGURE 13. INFLUENCES ON Z-POSITION. ERROR BARS SHOW 99% CI. (A) LOCATION (B) WINDS

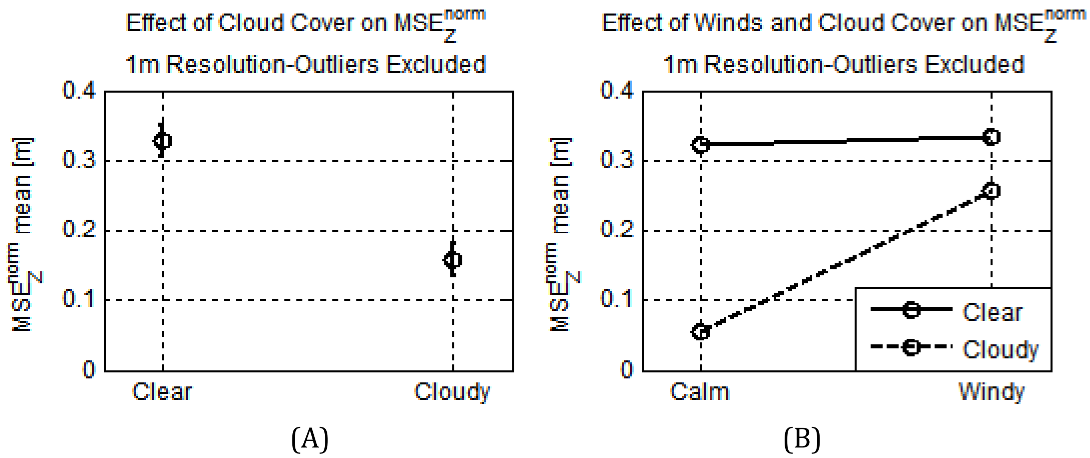


FIGURE 14. INFLUENCES ON Z-POSITION. ERROR BARS SHOW 99% CI. (A) CLOUD COVER (B) WINDS AND CLOUD COVER

Table 9 shows the planned contrast results for Z-position at 1m resolution. The same three independent variables as in Y-position are found to have significant differences for Z-position.

TABLE 9. PLANNED CONTRAST RESULTS FOR Z-POSITION AT 1M RESOLUTION

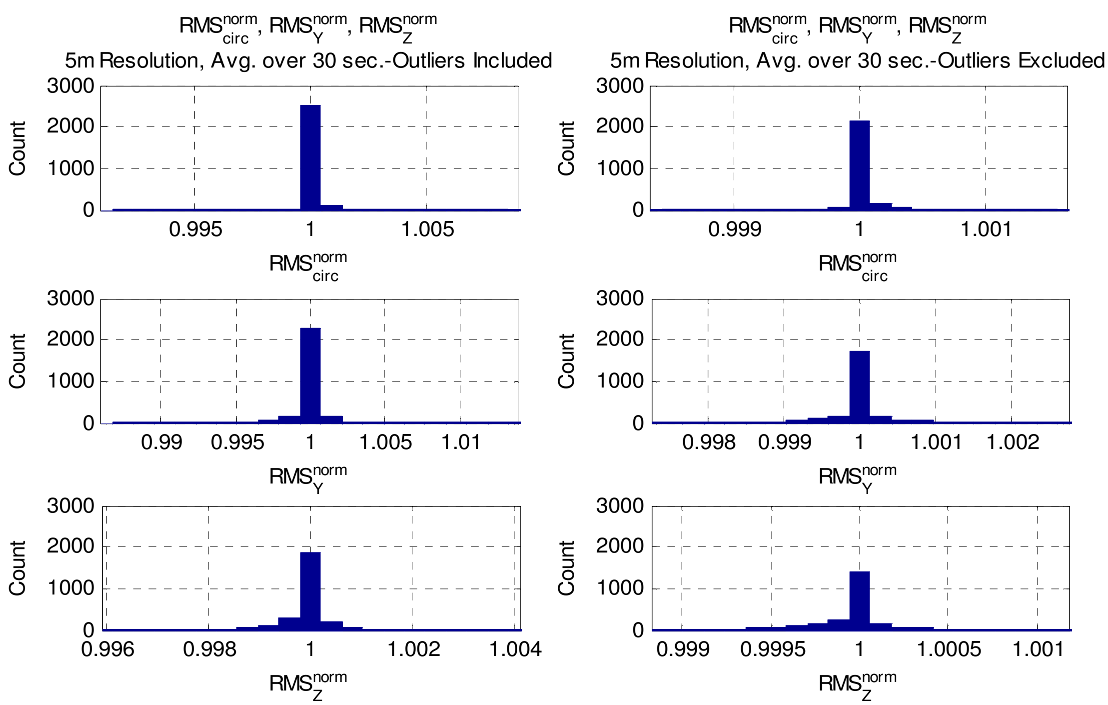
Group 1	Group 2	95% CI for True Mean	Significant?
APA	TDP	[-0.0632 0.0180]	No
DFW	MEM	[0.1504 0.1939]	Yes
Calm	Windy	[0.1504 0.1939]	Yes
Clear	Cloudy	[0.1237 0.2172]	Yes

COMPARISON OF PREDICTED DATA AT DIFFERENT INPUT RESOLUTIONS

DATA CHARACTERIZATION

In order to evaluate how the precision of the APA Suite output varies with input resolution, several analyses were done at several height resolutions, as detailed in the "Comparison of Predicted Data at Different Input Resolutions" section. In order to get a clear picture of the nature of the predicted data from the APA Suite, the normalized RMS values for circulation, Y-Position, and Z-Position were calculated, as discussed before, and plotted in histograms and box plots. This was done over the first 30 seconds of data as well as the first 60 seconds, in order to get ascertain how the data varied over time. The histogram and box plot parameters, including bin sizes, outlier limits, etc. are all identical to those in the "Comparison of Predicted and Measured Data" section.

To gain a general understanding of the effect of height resolution on the dependent variables, histograms and box plots for the 5, 40, and 100m resolutions are presented. Figure 15 and Figure 16 show the histograms for the 5m resolution averaged over 30 sec. and 60 sec. Judging by both the number of values in the central bins of the histograms before and after outliers are excluded, and by the box plots in Figure 17 and Figure 18, there are very few outliers at 5m resolution.



(A) (B)

FIGURE 15. HISTOGRAMS FOR 5M RESOLUTION, 30 SEC AVG (A) WITH OUTLIERS (B) WITHOUT OUTLIERS

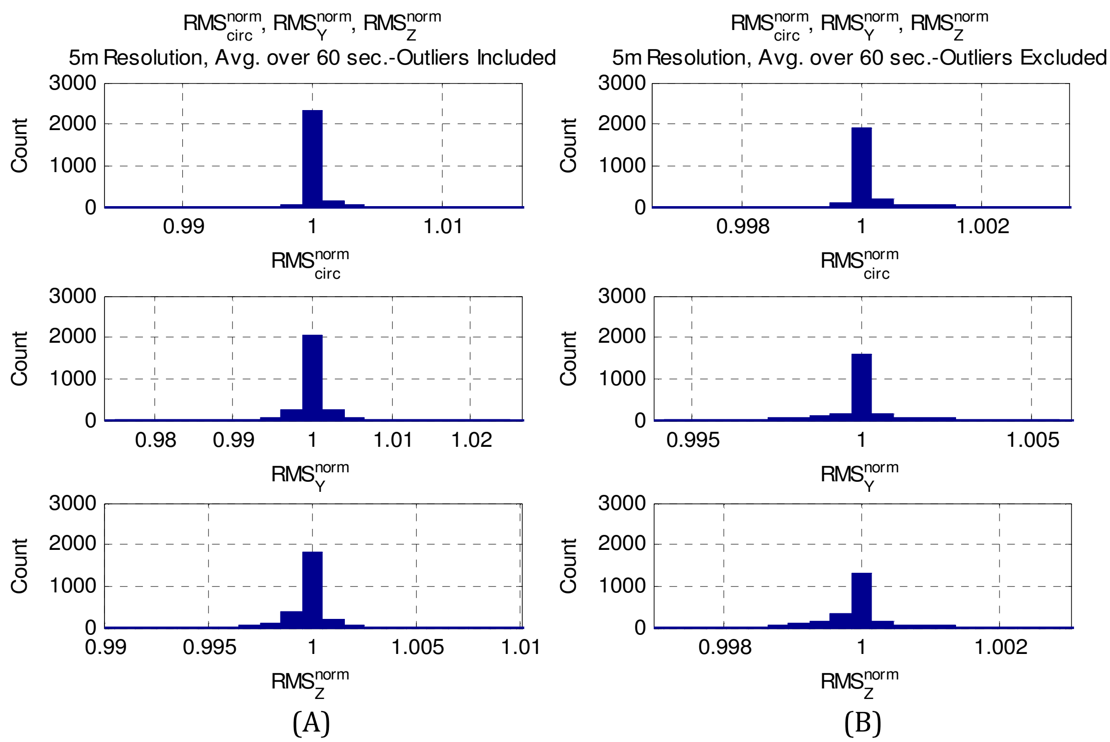


FIGURE 16. HISTOGRAMS FOR 5M RESOLUTION, 60 SEC AVG (A) WITH OUTLIERS (B) WITHOUT OUTLIERS

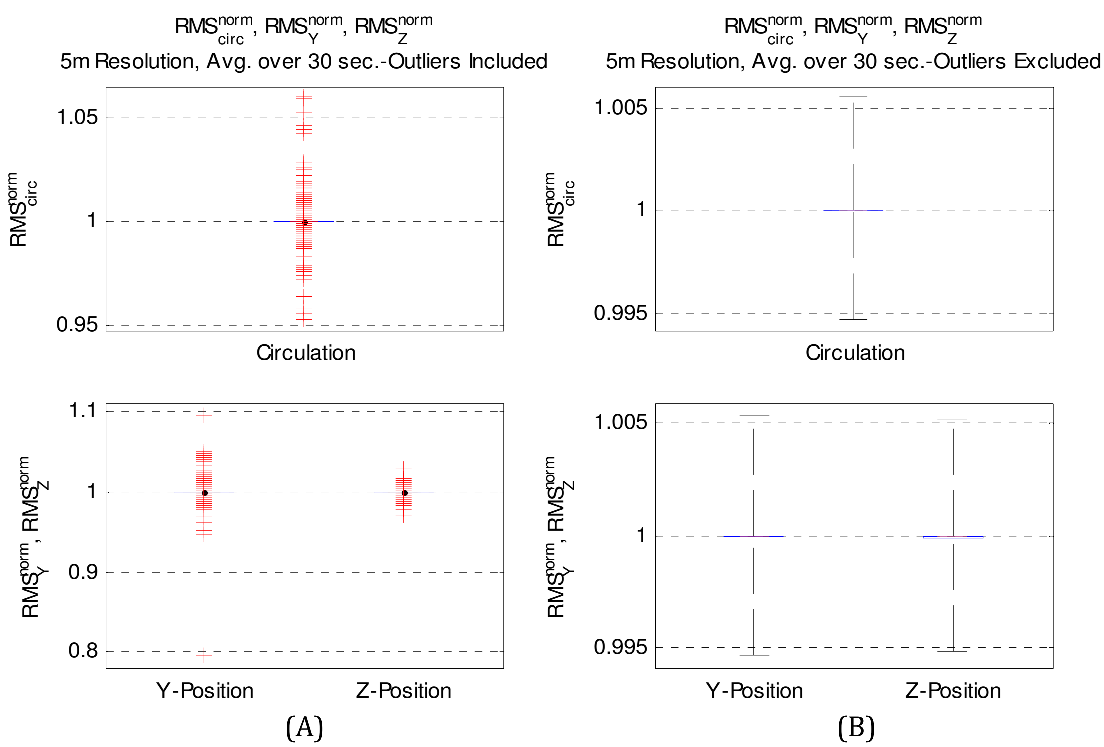


FIGURE 17. BOX PLOTS FOR 5M RESOLUTION, 30 SEC AVG (A) WITH OUTLIERS (B) WITHOUT OUTLIERS.

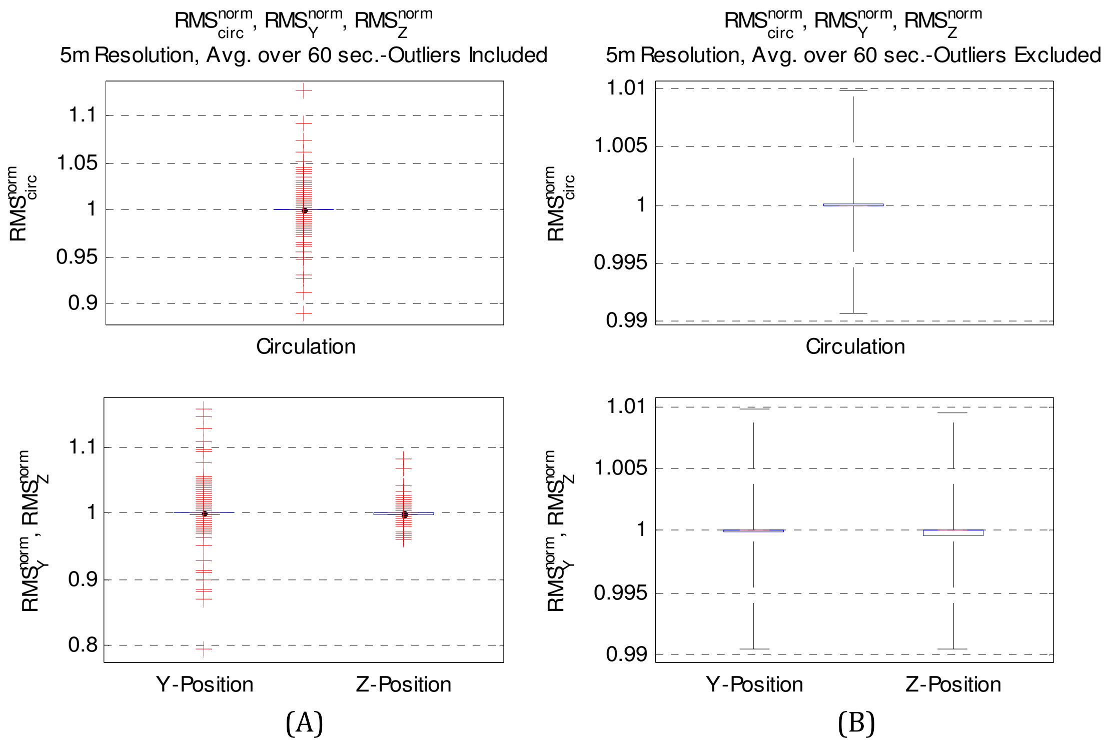


FIGURE 18. BOX PLOTS FOR 5M RESOLUTION, 60 SEC AVG (A) WITH OUTLIERS (B) WITHOUT OUTLIERS

Figure 19 and Figure 20 show the histograms for 20m resolution. Figure 21 and Figure 22 show the corresponding box plots. At 20m resolution there are more outliers than in 5m resolution, and thus the variance in the data is larger. The Z-position seems to be less affected by outliers than circulation and Y-position. However, at the 60 sec. average, all the dependent variables have a significant amount of outliers. This shows that the variance in the data from 1m resolution grows with both resolution and the time over which the average RMS is taken.

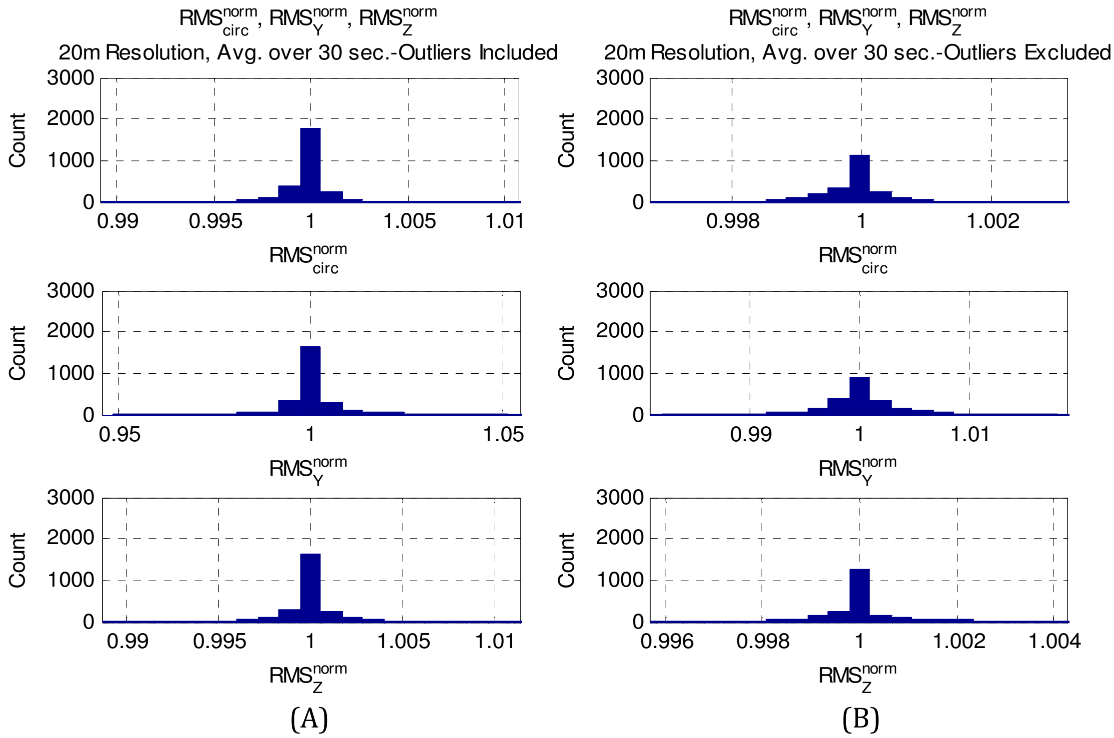


FIGURE 19. HISTOGRAMS FOR 20M RESOLUTION, 30 SEC AVG (A) WITH OUTLIERS (B) WITHOUT OUTLIERS

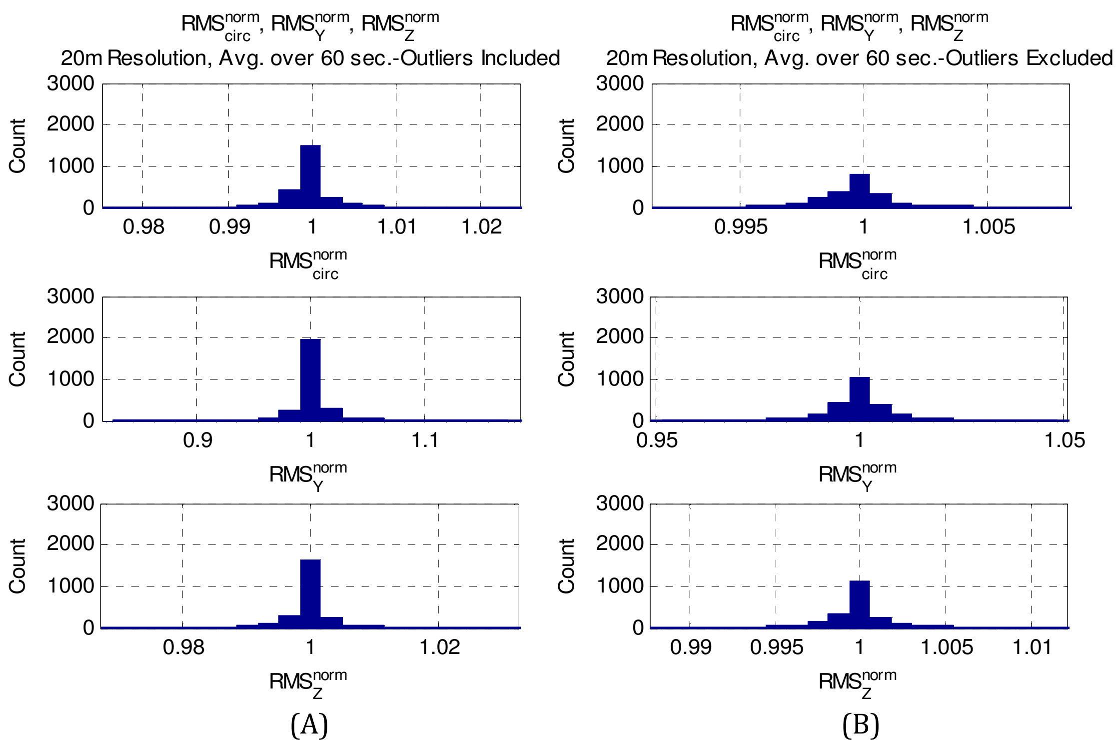
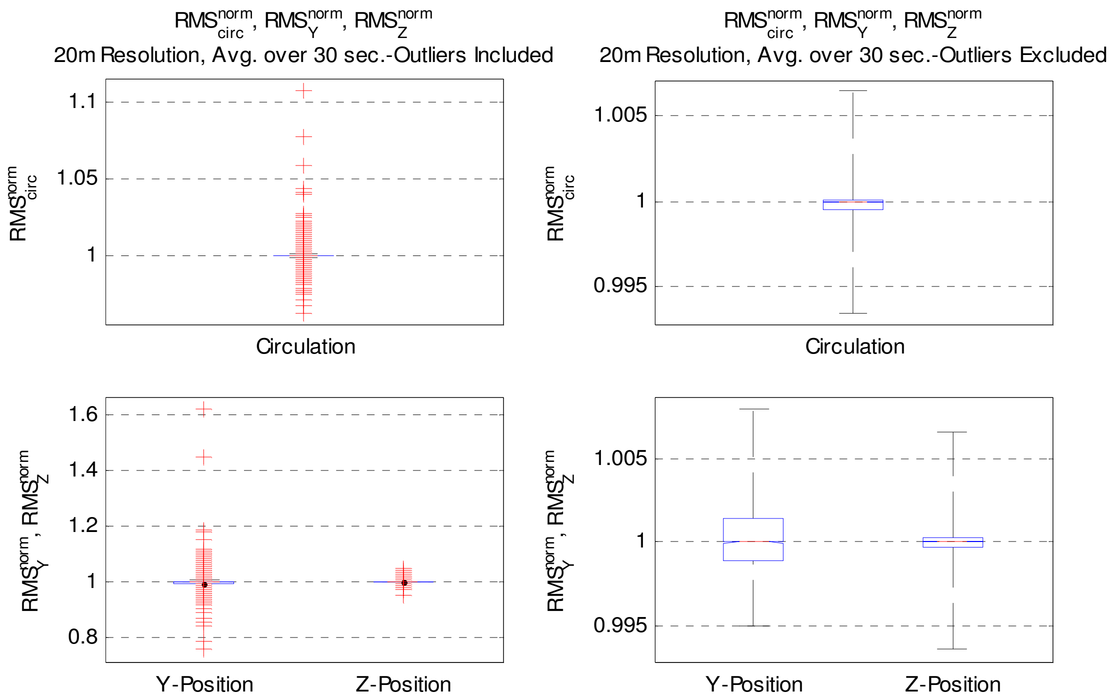
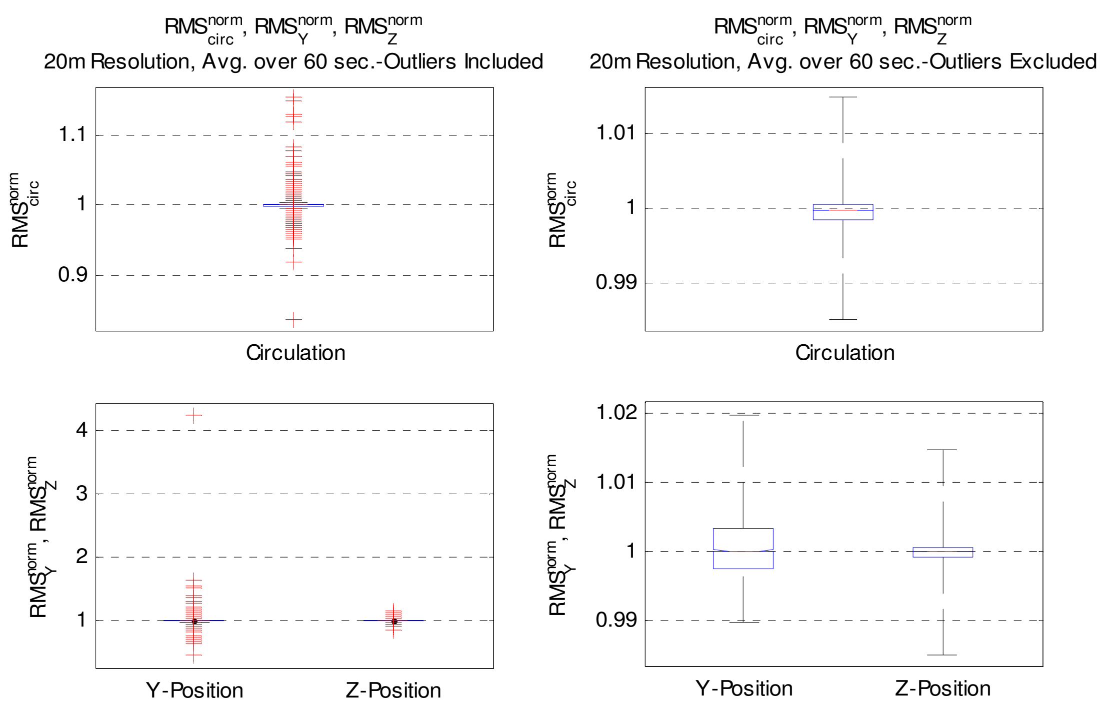


FIGURE 20. HISTOGRAMS FOR 20M RESOLUTION, 60 SEC AVG (A) WITH OUTLIERS (B) WITHOUT OUTLIERS.



(A) (B)

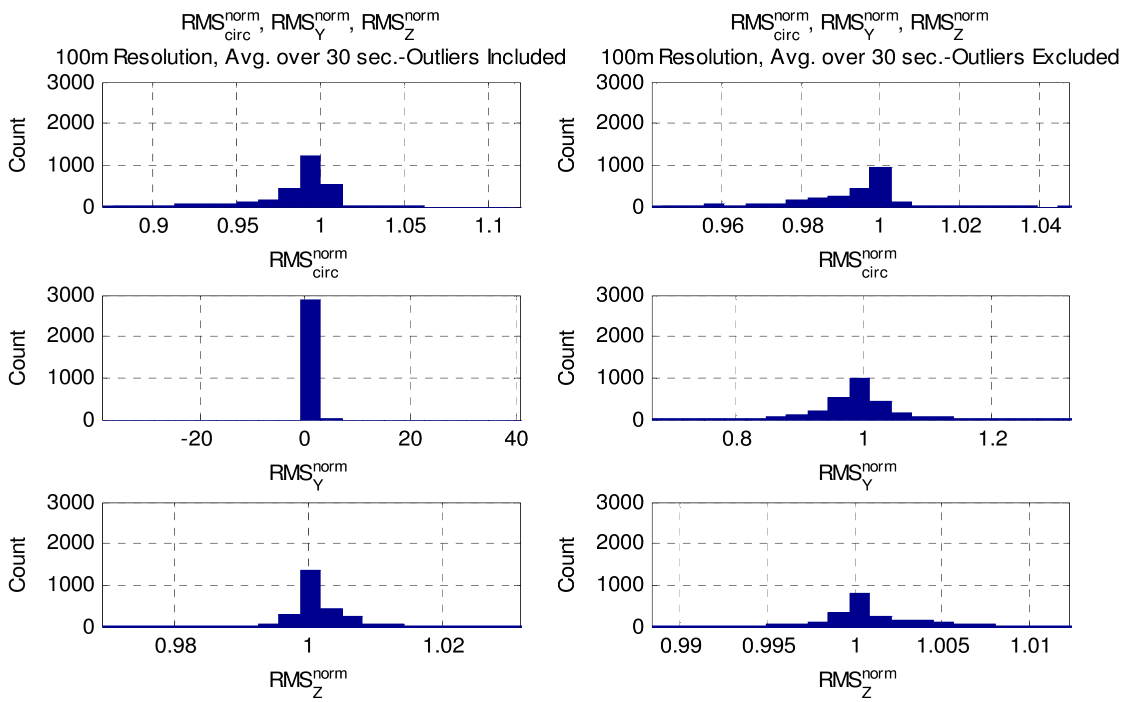
FIGURE 21. BOX PLOTS FOR 20M RESOLUTION, 30 SEC AVG (A) WITH OUTLIERS (B) WITHOUT OUTLIERS



(A) (B)

FIGURE 22. BOX PLOTS FOR 20M RESOLUTION, 60 SEC AVG (A) WITH OUTLIERS (B) WITHOUT OUTLIERS

Figure 23 and Figure 24 show histograms for 100m resolution. Here, there is a significant difference in the circulation with that for 1m resolution. At 100m resolution, the mean predicted circulation is significantly lower than that for 1m resolution, and the distribution is skewed to the left. The Y-position has a significantly higher variance than the circulation and Z-position in both the 30 sec. and 60 sec. averages. Figure 25 and Figure 26 show the box plots for this case. Precision is significantly degraded at 100m resolution, especially for Y-position. As will be later seen, 40m resolution (the next lower resolution), exhibits a distribution closer to that of the 20m resolution than the 100m resolution. To gain a clearer picture of at what resolution the precision of the dependent variables degrades the most, more resolutions should be run between the 40m and 100m values.



(A) (B)

FIGURE 23. HISTOGRAMS FOR 100M RESOLUTION, 30 SEC AVG (A) WITH OUTLIERS (B) W/O OUTLIERS

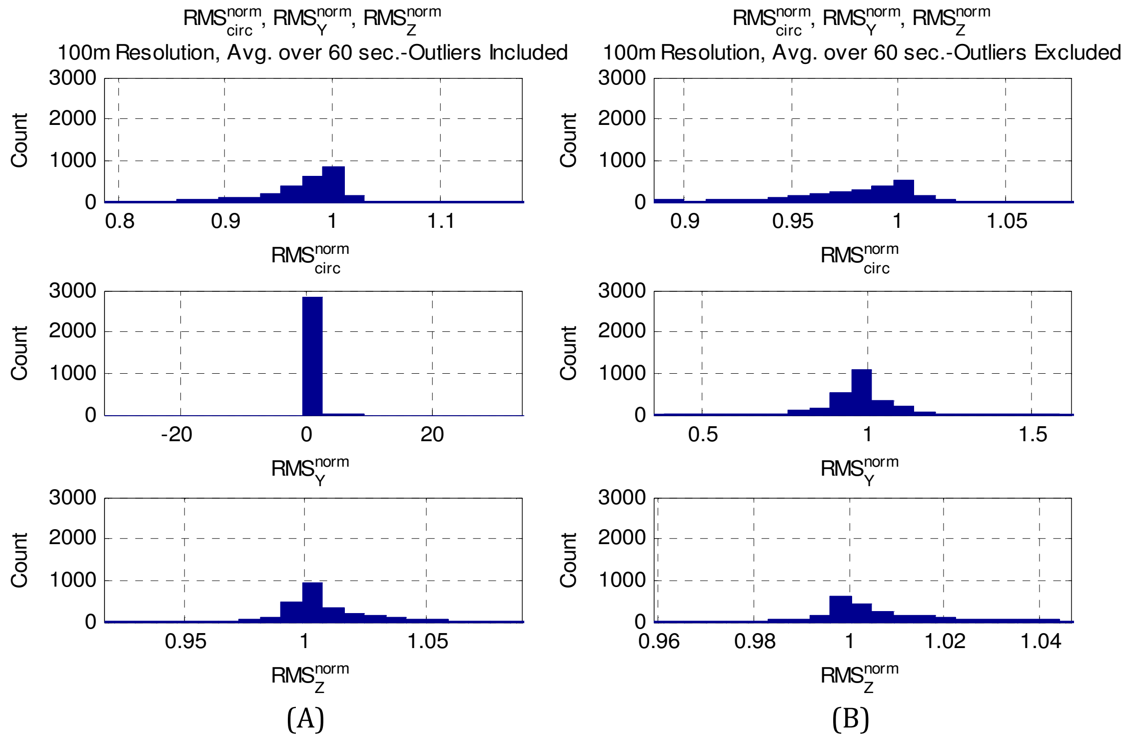


FIGURE 24. HISTOGRAMS FOR 100M RESOLUTION, 60 SEC AVG (A) WITH OUTLIERS (B) W/O OUTLIERS

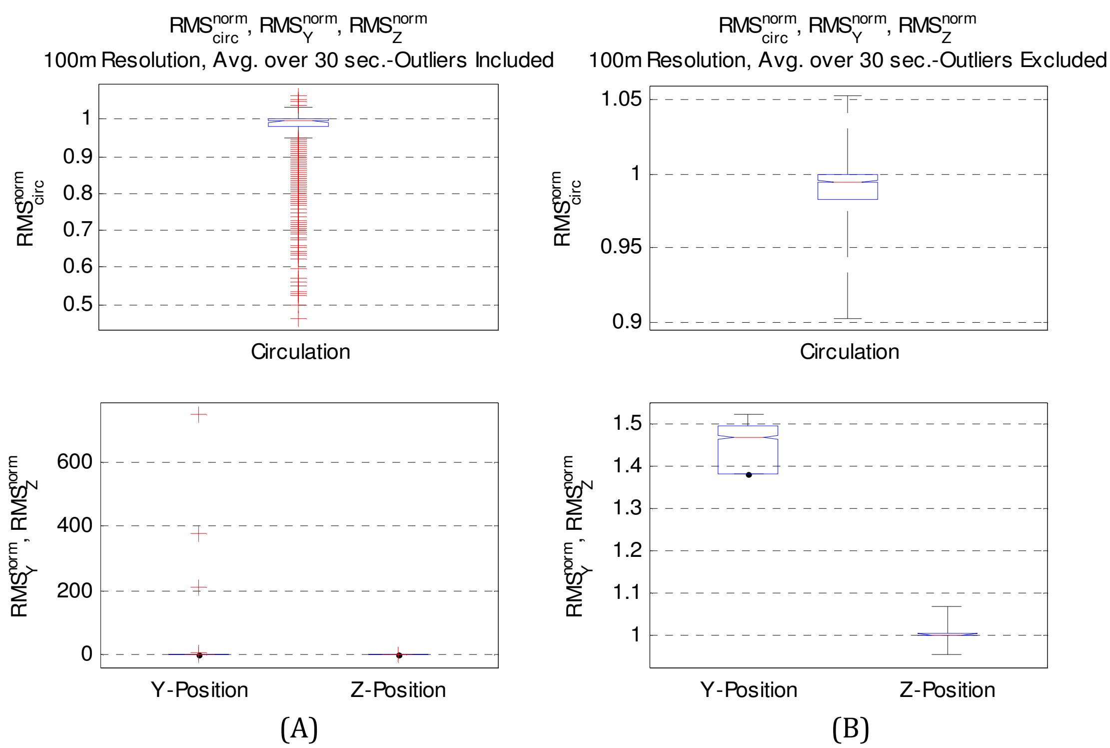


FIGURE 25. BOX PLOTS FOR 100M RESOLUTION, 30 SEC AVG (A) WITH OUTLIERS (B) WITHOUT OUTLIERS

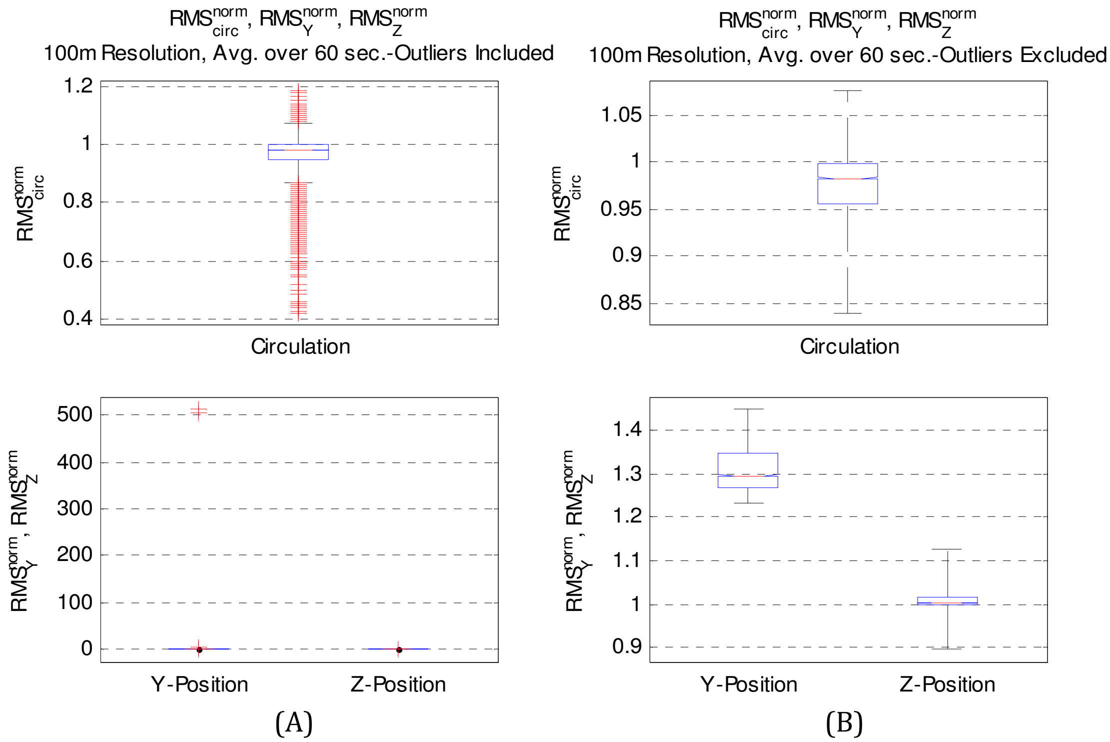


FIGURE 26. BOX PLOTS FOR 100M RESOLUTION, 60 SEC AVG (A) WITH OUTLIERS (B) WITHOUT OUTLIERS

OVERALL EFFECT OF INPUT DATA RESOLUTION ON PREDICTED DATA

An ANOVA analysis was run on each of the normalized RMS values of the dependent variables to ascertain the effect of height resolution on the data. Table 10 shows the result for circulation. The 5m to 25m resolutions do not play a significant role over the 30 sec. average, but the higher resolutions do (p-value < 0.01). This is an expected result, because as resolution increases, the precision of the output is expected to decrease. As the averaging time increases to 60 sec., the minimum resolution that plays a significant role decreases to 25m. Thus, over a larger time, the constraint on resolution should be tighter to ensure good precision in the data.

TABLE 10. RESOLUTION INFLUENCE ON CIRCULATION

Influences on Circulation	**30 sec. Average**		**60 sec. Average**	
	F	**Prob > F**	**F**	**Prob > F**
5m Resolution	0.17	0.6806	0.53	0.466
15m Resolution	0.78	0.3775	0.26	0.6134
20m Resolution	1.23	0.2668	3.61	0.0574
25m Resolution	2.86	0.0907	11.9	0.0006
30m Resolution	9.88	0.0017	23.04	0
40m Resolution	30	0	92.03	0
100m Resolution	2853.8	0	2342.22	0

Figure 27 shows a box plot of the circulation over all the resolutions and the two averaging times, with outliers removed. The maximum and minimum at each resolution is relatively the same, but the median changes significantly as resolution increases. Figure 28 is a zoomed in version of Figure 27, making the median at each resolution more clear. Based on the notches, signifying the 95% confidence interval of the population median, it is clear that this change in median is significant. As resolution increases, precision decreases, and the APA Suite under-predicts circulation as compared to the predicted values in 1m resolution.

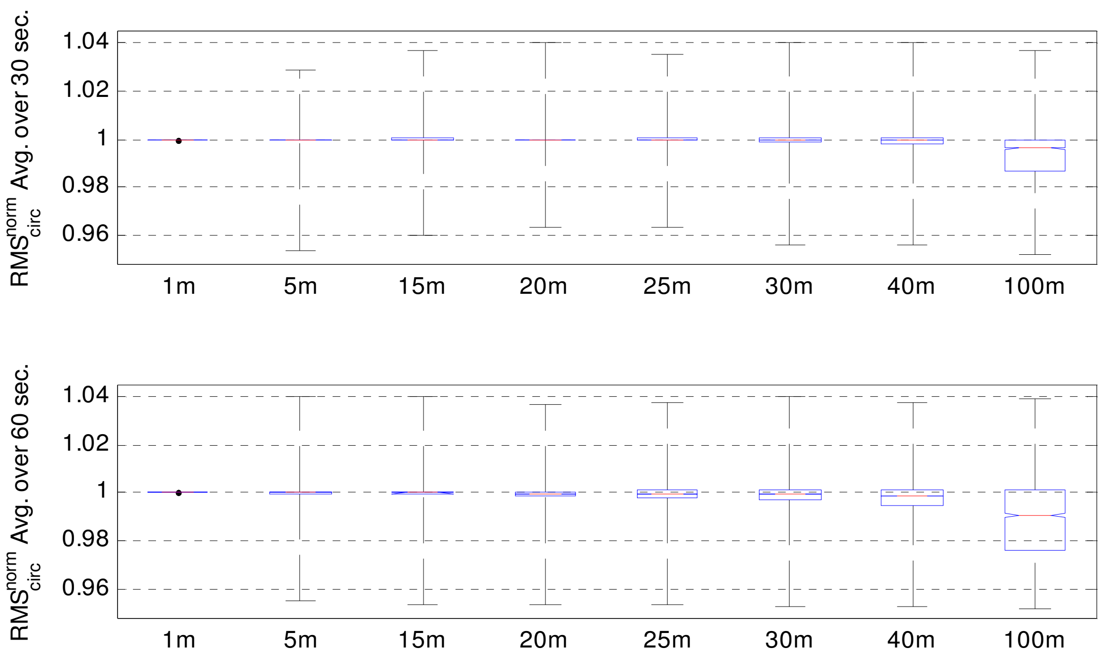


FIGURE 27. BOX PLOTS OF CIRCULATION FOR ALL HEIGHT RESOLUTIONS

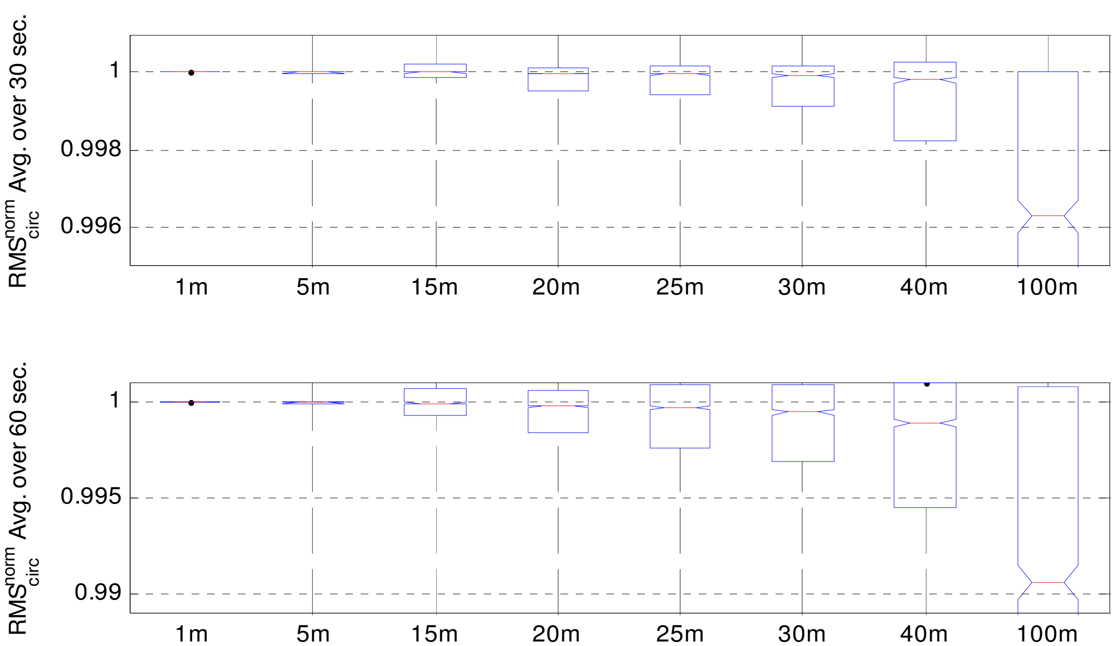


FIGURE 28. BOX PLOTS OF CIRCULATION FOR ALL HEIGHT RESOLUTIONS (ZOOMED)

Table 11 shows the ANOVA results for Y-position. Here, the resolution begins to play a significant role at 30m. However, at 40m, the effect is insignificant, and at 100m the effect is once again significant. The reason this occurs is unclear, but by looking at the corresponding box plots in Figure 29 and Figure 30, it is apparent that the median of the Y-position increases from 25m to 30m, and then decreases from 30m to 40m, going against the general trend.

TABLE 11. RESOLUTION INFLUENCE ON Y-POSITION

Influences on Y-Position	**30 sec. Average**		**60 sec. Average**	
	F	**Prob > F**	**F**	**Prob > F**
5m Resolution	0	0.9699	0	0.9466
15m Resolution	0.89	0.3462	1.62	0.203
20m Resolution	1.07	0.3	2.33	0.1269
25m Resolution	0.53	0.4666	0.35	0.5523
30m Resolution	9.14	0.0025	5.34	0.0209
40m Resolution	0.05	0.8149	0.88	0.3483
100m Resolution	57.9	0	7.71	0.0055

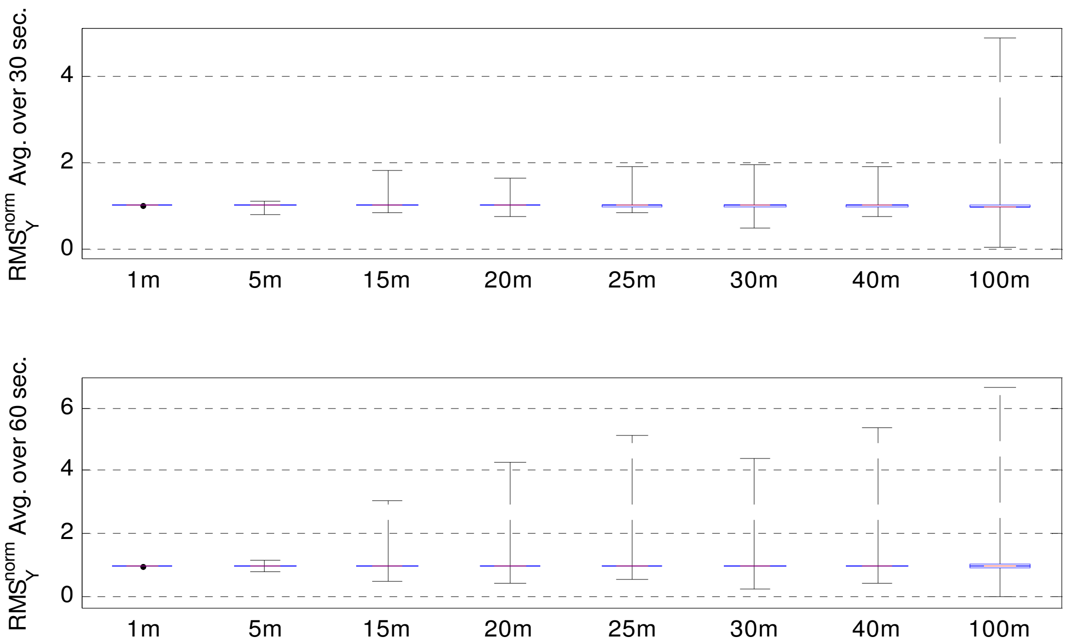


FIGURE 29. BOX PLOTS OF Y-POSITION FOR ALL HEIGHT RESOLUTIONS

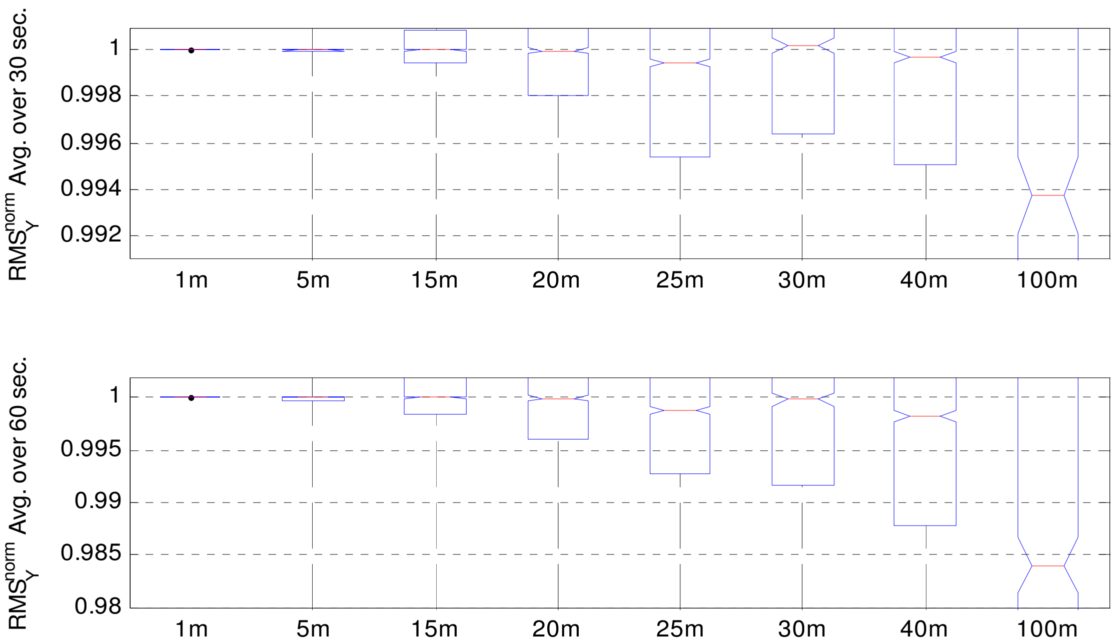


FIGURE 30. BOX PLOTS OF Y-POSITION FOR ALL HEIGHT RESOLUTIONS (ZOOMED)

Table 12 shows the ANOVA results for Z-position. Here, only the 100m resolution shows a significant role affecting the data. This can again be seen in Figure 31 and Figure 32, as 100m resolution is the only resolution that has an apparent change in median value. This shows that Z-position prediction is very robust to resolution changes. However as the resolution increases, the Z-position is over-predicted, not under-predicted as before.

TABLE 12. RESOLUTION INFLUENCE ON Z-POSITION

Influences on Z-Position	**30 sec. Average**		**60 sec. Average**	
	F	**Prob > F**	**F**	**Prob > F**
5m Resolution	1.2	0.2739	3.17	0.0752
15m Resolution	0	0.9712	2.87	0.0903
20m Resolution	0.6	0.4371	0.77	0.379
25m Resolution	0.01	0.9265	1.22	0.2688
30m Resolution	0.04	0.8494	0.23	0.6342
40m Resolution	1.96	0.1617	1.58	0.2084
100m Resolution	178.77	0	205.99	0

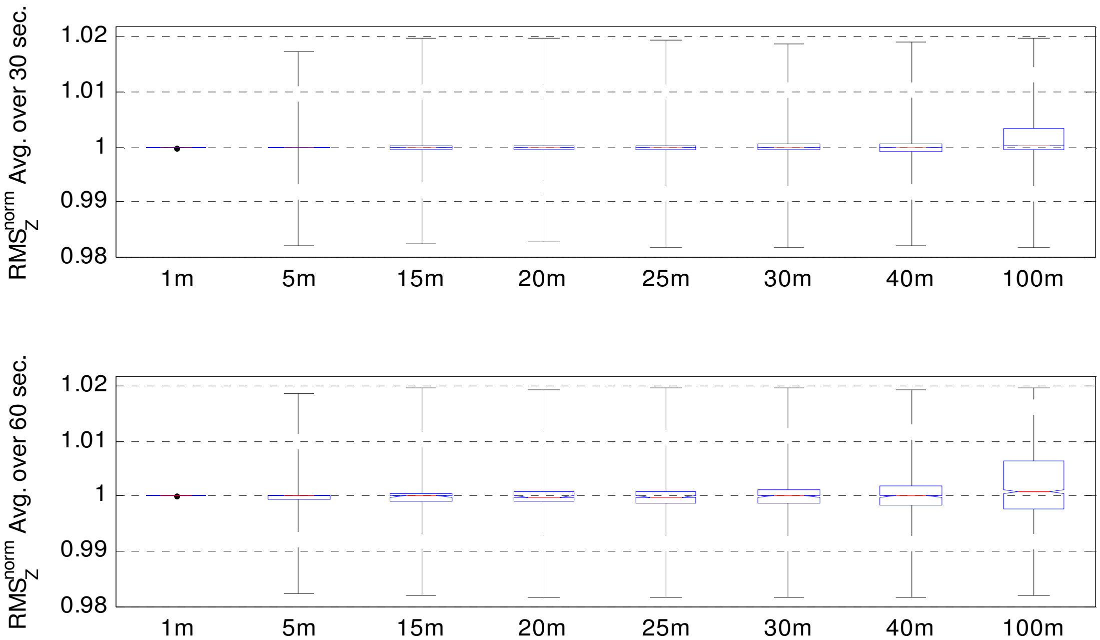


FIGURE 31. BOX PLOTS OF Z-POSITION FOR ALL HEIGHT RESOLUTIONS

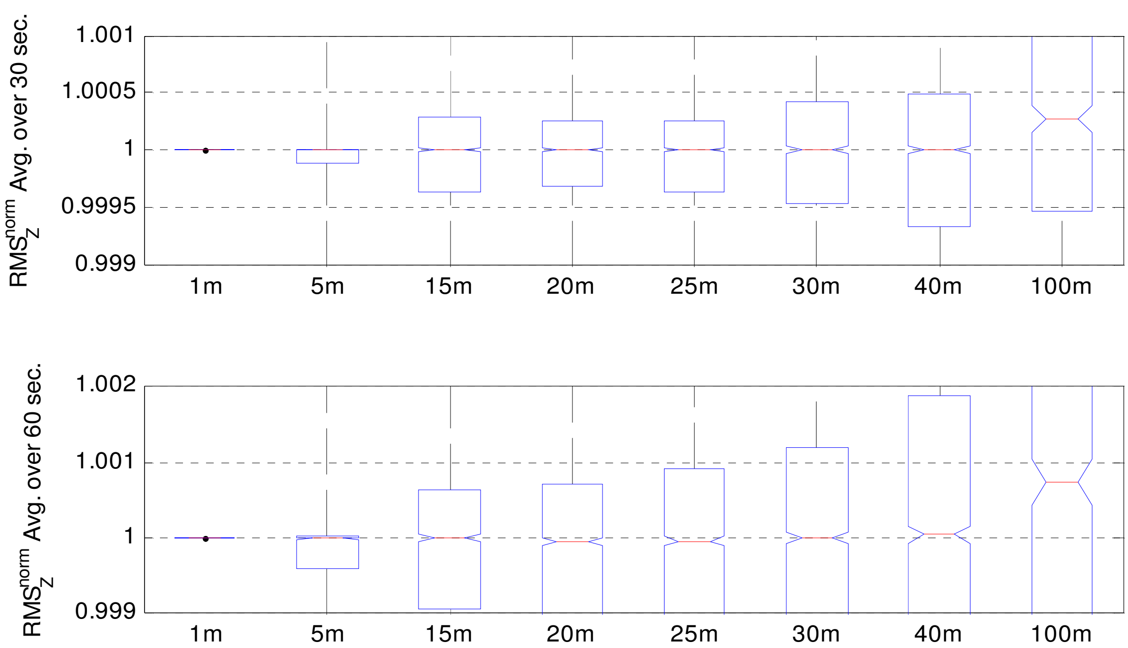


FIGURE 32. BOX PLOTS OF Z-POSITION FOR ALL HEIGHT RESOLUTIONS (ZOOMED)

To further determine the effects of the independent variables on circulation, an ANOVA was performed on the normalized circulation RMS with all of the independent variables. The result is shown in Table 13, and all of the independent variables play a role in the precision of the circulation.

TABLE 13. INFLUENCES OF INDEPENDENT VARIABLES ON CIRCULATION

Influences on Circulation	**30 sec. Average**		**60 sec. Average**	
	F	**Prob > F**	**F**	**Prob > F**
Analysis Model (APA or TDP)	65.15	0	12.98	0.0003
Location (DFW or MEM)	208	0	208.06	0
Aircraft Weight Class (Heavy, Large, or Small)	33.7	0	17.43	0
Winds (Windy or Calm)	3.4	0.0651	86.44	0
Cloud Cover (Cloudy or Clear)	21.65	0	9.93	0.0016
Resolution (1m, 5m, ..., 100m)	498.35	0	401.35	0
Analysis Model and Winds	6.94	0.0084	10.79	0.001
Analysis Model and Cloud Cover	0.11	0.7348	13.16	0.0003
Winds and Cloud Cover	1.77	0.1838	33.83	0

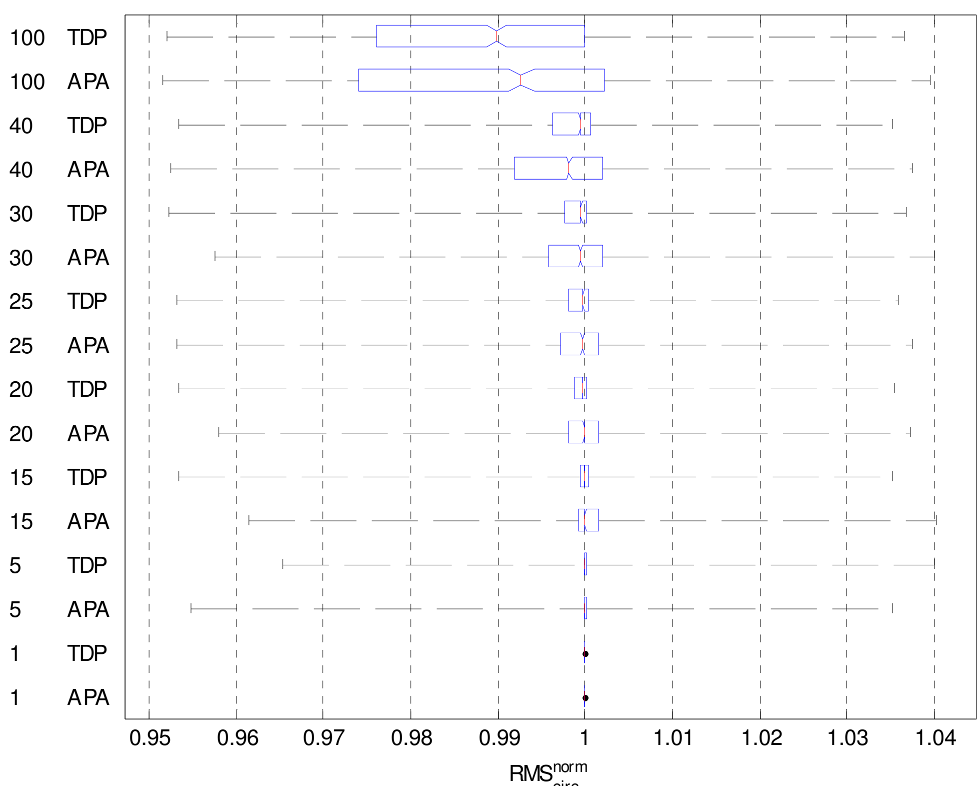


FIGURE 33. EFFECT OF RESOLUTION AND ANALYSIS TYPE ON CIRCULATION

Since the 60 sec. average produces the more conservative estimate for precision, only these values are used in the following box plots. Figure 33 shows a box plot of the normalized circulation RMS as compared to resolution and analysis type. At high resolution, there is essentially no difference between the two analysis types. However, as resolution is reduced, the TDP analysis method appears to have a lower variance and a median that is closer to 1.0 than the APA method. At the very low resolution of 100m, both methods show a very large offset from the 1m median.

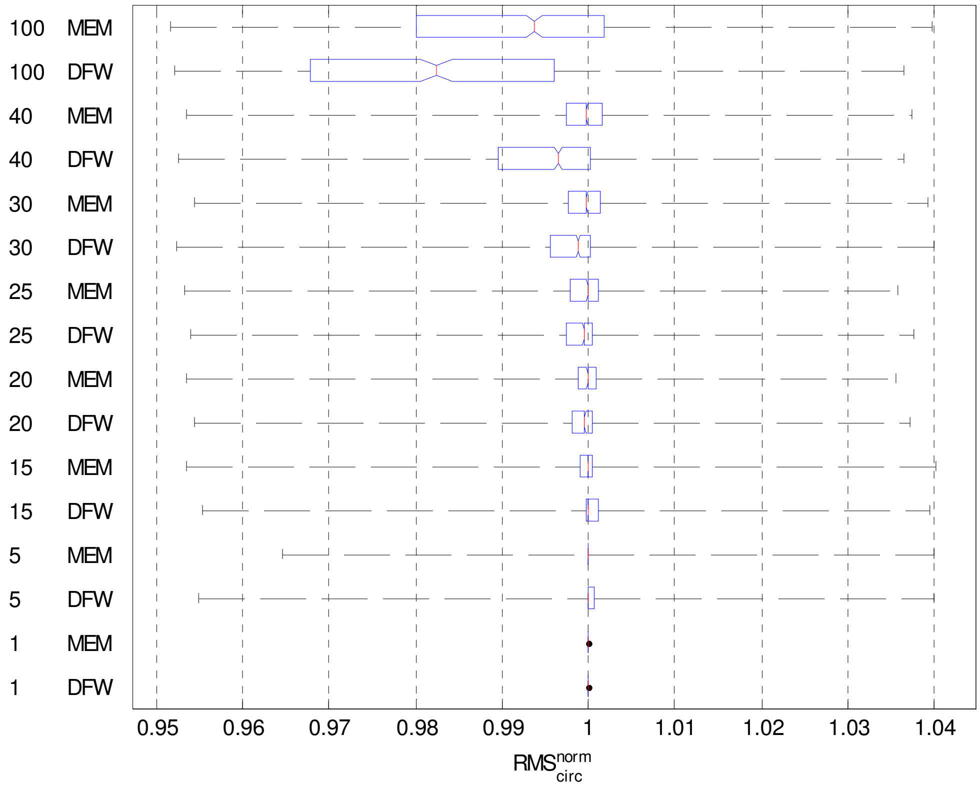


FIGURE 34. EFFECT OF RESOLUTION AND LOCATION ON CIRCULATION

Figure 34 shows a box plot of circulation as before, but comparing resolution and location instead. MEM produced much more precise results than DFW. This can only be due to the velocity profiles provided for each location. Otherwise, the location should not make a difference in terms of the precision versus resolution aspect of the model.

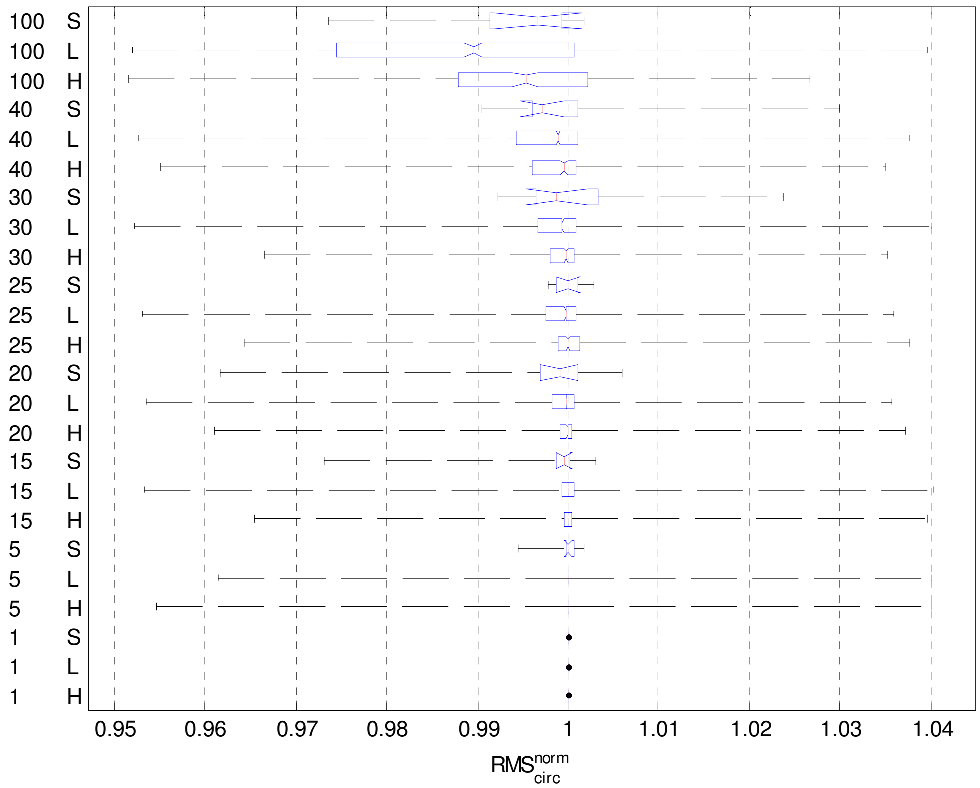


FIGURE 35. EFFECT OF RESOLUTION AND AIRCRAFT WEIGHT CLASS ON CIRCULATION (S=SMALL, L=LARGE, H=HEAVY)

Figure 35 shows a box plot of the circulation compared to resolution and aircraft weight class. Generally, at high resolution there is very little difference among aircraft classes. Small aircraft seem to show the largest variance and offset from 1.0, and thus show the greatest change in precision with resolution. At lower resolutions, this effect becomes more pronounced, and large aircraft exhibit the same effect, although to a lesser degree.

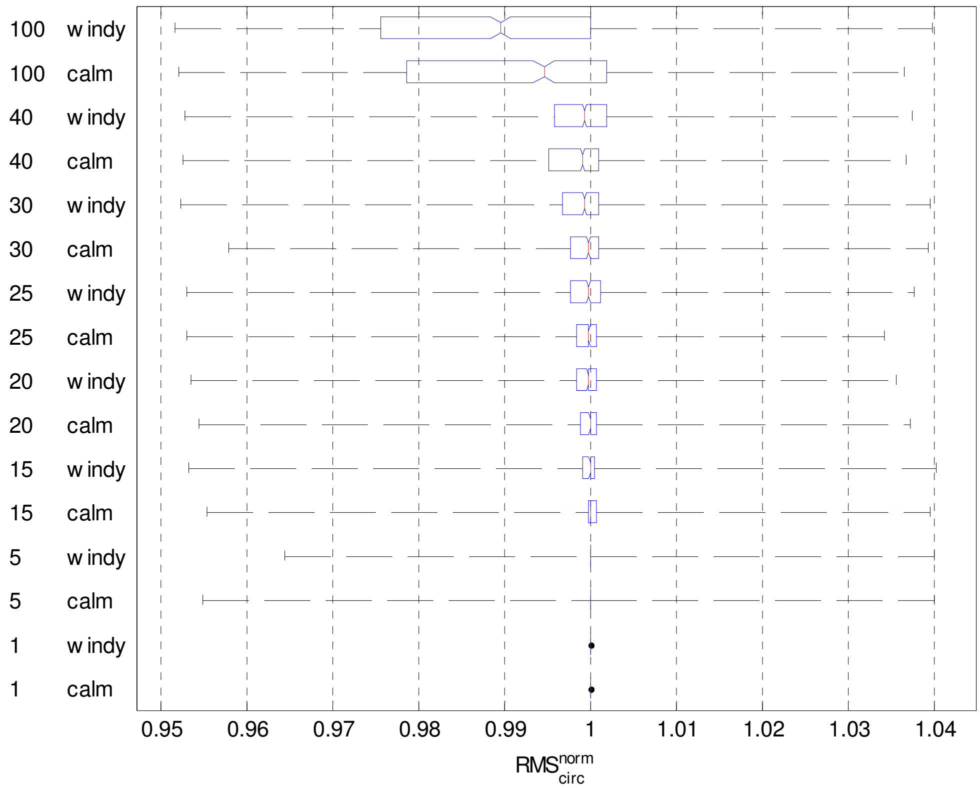


FIGURE 36. EFFECT OF RESOLUTION AND WIND CONDITION ON CIRCULATION

Figure 36 shows a box plot of the circulation versus resolution and wind condition. Calm winds generally indicate a lower variance from the 1m resolution, while windy conditions indicate a larger variance and larger offset from the 1m resolution. This might speak to the accuracy of the wind model at high wind speeds in the APA Suite.

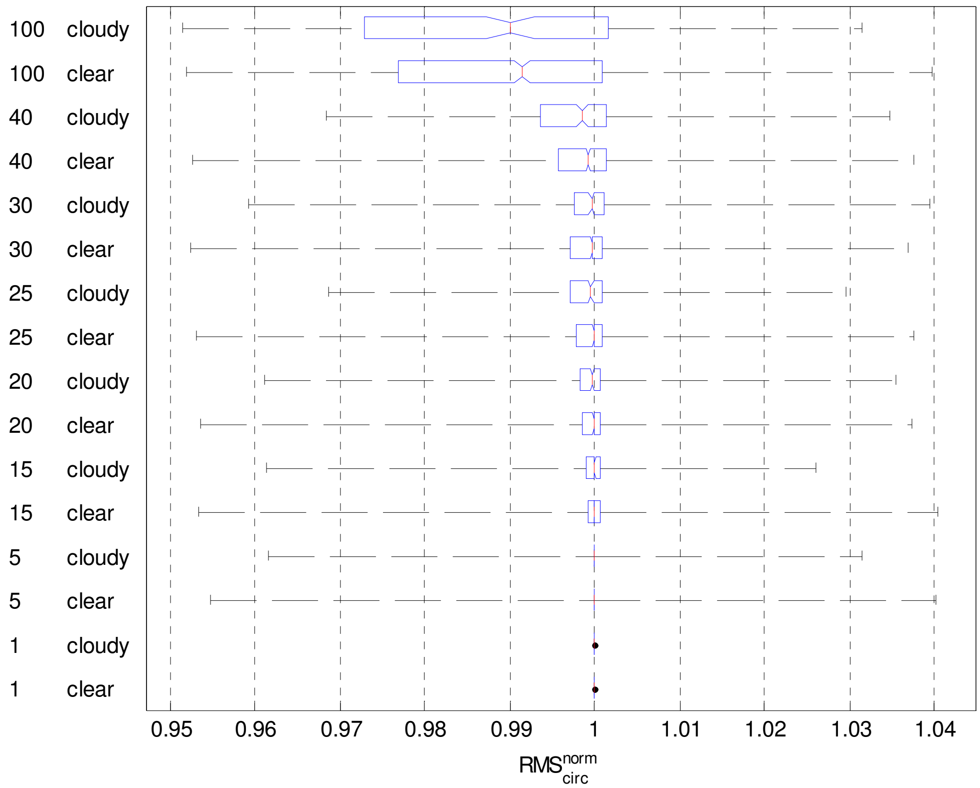


FIGURE 37. EFFECT OF RESOLUTION AND CLOUD COVER ON CIRCULATION

Figure 37 shows the effect of resolution and cloud cover on circulation. Cloudy conditions seem to produce a higher variance than clear conditions, and this effect increases as resolution decreases. Note that overall, in all the box plots for circulation, the normalized RMS always remains with 5% of the RMS of the 1m resolution, so these perceived results of bad precision are not very large.

Table 14 shows the planned contrast results for circulation. Analysis type, location, winds, and cloud conditions all play a significant role in circulation precision.

TABLE 14. PLANNED CONTRAST RESULTS FOR CIRCULATION

	Group 1	Group 2	95% CI for True Mean	Significant?
30 sec. Average	APA	TDP	[-7.3521E-4 -4.4786E-4]	Yes
	DFW	MEM	[-0.0013 -0.0010]	Yes
	Calm	Windy	[-9.0389E-6 2.9748E-4]	No
	Clear	Cloudy	[3.0179E-4 7.4112E-4]	Yes
	Group 1	**Group 2**	**95% CI for True Mean**	**Significant?**
60 sec. Average	APA	TDP	[1.9368E-4 6.5563E-4]	Yes
	DFW	MEM	[-0.0022 -0.0016]	Yes
	Calm	Windy	[9.2441E-4 0.0014]	Yes
	Clear	Cloudy	[-9.2223E-4 -2.1495E-4]	Yes

ANOVA results for the normalized Y-position RMS are shown in Table 15. These results show that the significant contributors to Y-position precision are analysis type, location, and aircraft weight class. Cloud cover may play a role in the short term (30 sec. average), but diminishes in the long term (60 sec. average).

TABLE 15. INFLUENCES OF INDEPENDENT VARIABLES ON Y-POSITION

Influences on Y-Position	30 sec. Average		60 sec. Average	
	F	Prob > F	F	Prob > F
Analysis Model (APA or TDP)	1.81	0.1785	10.37	0.0013
Location (DFW or MEM)	31.74	0	32.06	0
Aircraft Weight Class (Heavy, Large, or Small)	14.27	0	5.33	0.0048
Winds (Windy or Calm)	0.59	0.4435	2.34	0.1262
Cloud Cover (Cloudy or Clear)	3.8	0.0514	0.7	0.4023
Resolution (1m, 5m, ..., 100m)	14.47	0	2.2	0.0311
Analysis Model and Winds	1.59	0.2073	11.78	0.0006
Analysis Model and Cloud Cover	4.2	0.0404	37.78	0
Winds and Cloud Cover	0.1	0.7523	5.36	0.0206

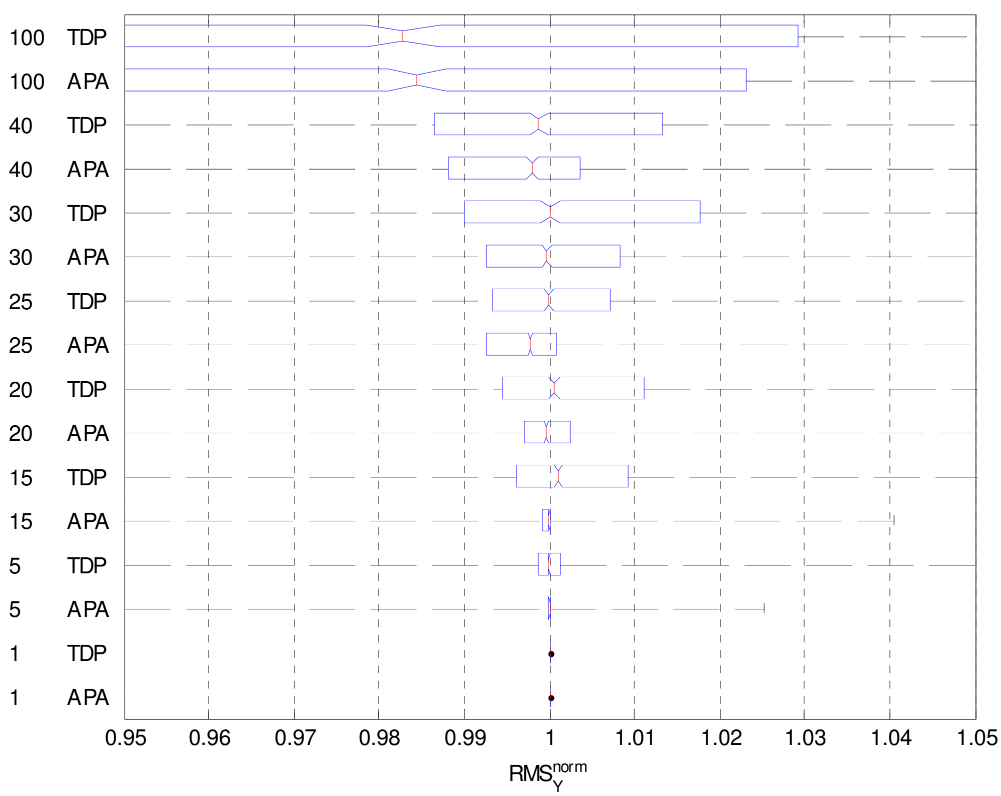


FIGURE 38. EFFECT OF RESOLUTION AND ANALYSIS TYPE ON Y-POSITION

Figure 38 shows a box plot comparing the effects of resolution and analysis type on Y-position precision. The median values of the TDP method match that of the 1m resolution more closely than the APA method, but the TDP method exhibits a larger variance than the APA method. Note that in this and all following box plots for Y-position, the minimums and maximums always extend beyond 5% of the 1m resolution, and at the resolution of 100m, the middle 50% of the data also extends beyond this 5% boundary.

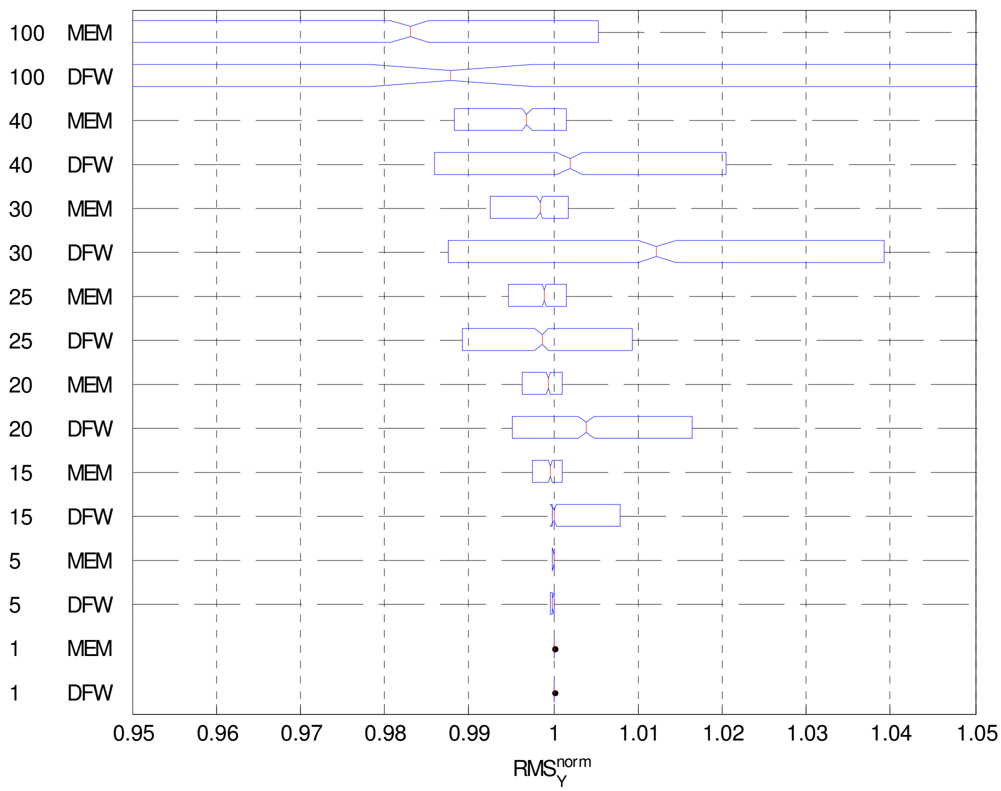


FIGURE 39. EFFECT OF RESOLUTION AND LOCATION ON Y-POSITION

Figure 39 shows the same data but compared to resolution and location. Again, MEM appears to produce results that both have lower variance and are closer to the 1m resolution data than DFW.

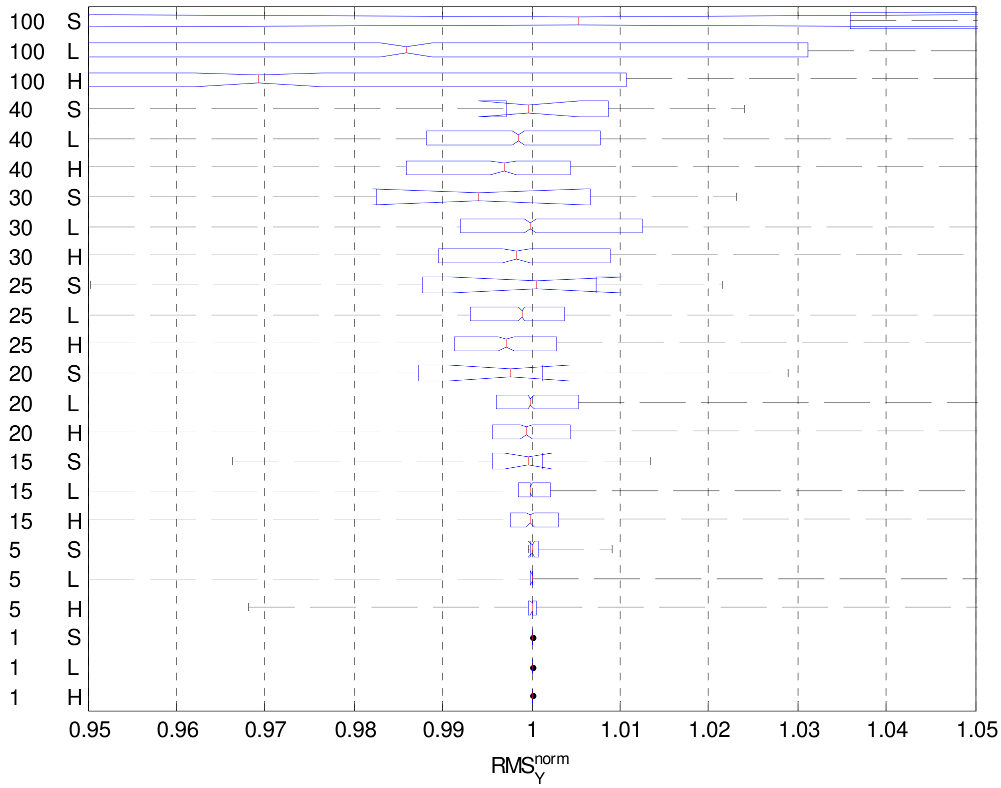


FIGURE 40. EFFECT OF RESOLUTION AND AIRCRAFT WEIGHT CLASS ON Y-POSITION (S=SMALL, L=LARGE, H=HEAVY)

Figure 40 shows the comparison of Y-position precision and aircraft weight class. “Small” aircraft produce a large variance and offset from the 1m resolution, and “heavy” aircraft produce a much smaller variance and offset, at all resolutions.

Table 16 shows the planned contrast results for Y-position precision. Analysis type and location play a significant role, while winds and cloud cover do not.

TABLE 16. PLANNED CONTRAST RESULTS FOR Y-POSITION

	Group 1	**Group 2**	**95% CI for True Mean**	**Significant?**
30 sec. Average	APA	TDP	[-0.0024 4.5506E-4]	No
	DFW	MEM	[0.0030 0.0062]	Yes
	Calm	Windy	[-9.4422E-4 0.0022]	No
	Clear	Cloudy	[-0.0044 1.3500E-5]	No
60 sec. Average	**Group 1**	**Group 2**	**95% CI for True Mean**	**Significant?**
	APA	TDP	[-0.008 -0.0021]	Yes
	DFW	MEM	[0.0070 0.0143]	Yes
	Calm	Windy	[-7.8089E-4 0.0063]	No
	Clear	Cloudy	[-0.0073 0.0029]	No

Table 17 shows the ANOVA results on the factors affecting Z-position precision. From this data, analysis type, location, aircraft weight class, and (only at the 30 sec. average) cloud cover play a significant role in Z-position precision. The following box plots will illustrate the effects these independent variables produce.

TABLE 17. INFLUENCES OF INDEPENDENT VARIABLES ON Z-POSITION

Influences on Z-Position	**30 sec. Average**		**60 sec. Average**	
	F	**Prob > F**	**F**	**Prob > F**
Analysis Model (APA or TDP)	23	0	44.91	0
Location (DFW or MEM)	41.47	0	82.12	0
Aircraft Weight Class (Heavy, Large, or Small)	1.39	0.2493	14.19	0
Winds (Windy or Calm)	0.04	0.8497	2.01	0.1558
Cloud Cover (Cloudy or Clear)	7.49	0.0062	0.51	0.4759
Resolution (1m, 5m, ..., 100m)	22.37	0	41.18	0
Analysis Model and Winds	9.05	0.0026	3.54	0.06
Analysis Model and Cloud Cover	0.74	0.3892	0.53	0.466
Winds and Cloud Cover	29.46	0	57.43	0

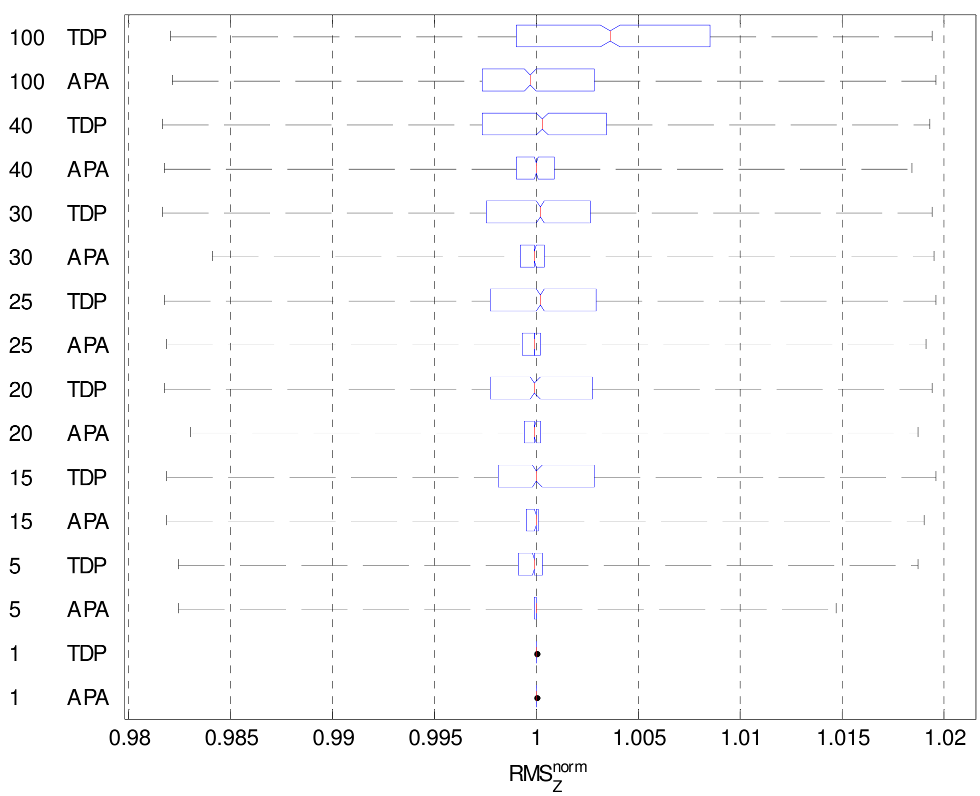


FIGURE 41. EFFECT OF RESOLUTION AND ANALYSIS TYPE ON Z-POSITION

Figure 41 shows the effect of resolution and analysis type on Z-position precision. The APA method shows a significantly lower variance than the TDP method, but both methods remain close to the 1m resolution for most of the resolutions. However, at 100m resolution, the APA method remains close to the 1m median while the TDP drifts approximately 0.4% from the 1m median. All things considered, this drift is very minor, and thus either method is suitable for predicting Z-position at all resolutions.

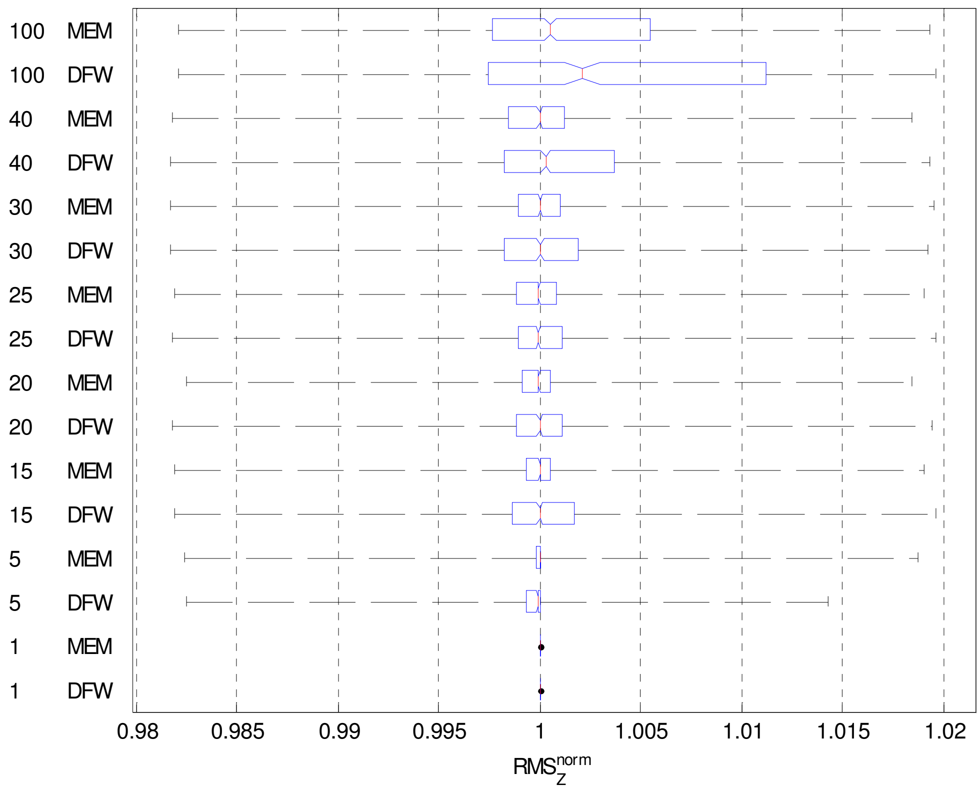


FIGURE 42. EFFECT OF RESOLUTION AND LOCATION ON Z-POSITION

Figure 42 shows the effect of resolution and location on Z-position precision. Again, the effects are very minor, with both locations showing approximately the same variance from the 1m median. The variance does grow as resolution decreases, but the range of all the data still remains well within 5% of the 1m resolution data.

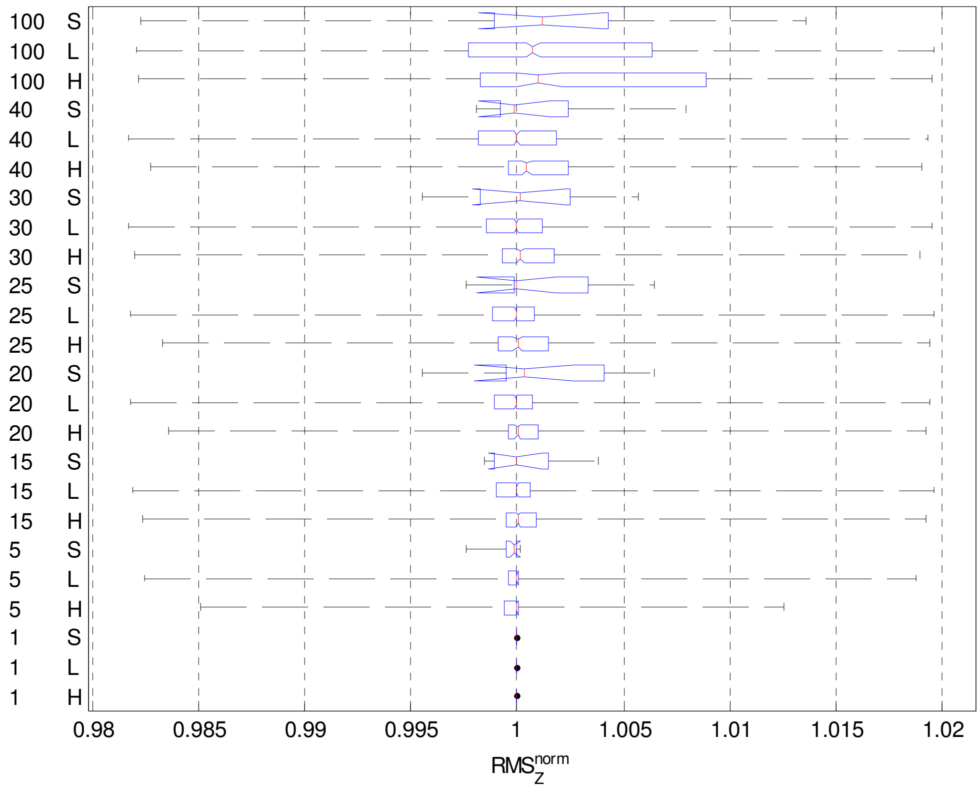


FIGURE 43. EFFECT OF RESOLUTION AND AIRCRAFT WEIGHT CLASS ON Z-POSITION (S=SMALL, L=LARGE, H=HEAVY)

Figure 43 shows the effect of aircraft weight class and resolution on Z-position precision. In this case, "small" aircraft actually produce a *smaller* variance than the other aircraft types, at all resolutions. Overall, the percent errors are still much less than 5% of the 1m resolution data, so aircraft weight class does not drastically affect Z-position precision. Table 18 shows the planned contrast results for Z-position. These results confirm the results shown in the ANOVA table.

TABLE 18. PLANNED CONTRAST RESULTS FOR Z-POSITION

	Group 1	**Group 2**	**95% CI for True Mean**	**Significant?**
30 sec. Average	APA	TDP	[1.2406E-4 2.9553E-4]	Yes
	DFW	MEM	[2.1740E-4 4.0762E-4]	Yes
	Calm	Windy	[-8.2750E-5 1.0046E-4]	No
	Clear	Cloudy	[-3.1544E-4 -5.2192E-5]	Yes
60 sec. Average	**Group 1**	**Group 2**	**95% CI for True Mean**	**Significant?**
	APA	TDP	[-5.7211E-4 -3.1319E-4]	Yes
	DFW	MEM	[5.2886E-4 8.2076E-4]	Yes
	Calm	Windy	[-2.3822E-4 3.8111E-5]	No
	Clear	Cloudy	[-2.6880E-4 1.2540E-4]	No

SUMMARY

A sensitivity analysis of the APA Suite has been conducted and the data analyzed in two parts. The first sought to characterize the accuracy of the APA Suite predictions for 1m (highest) and 100m (lowest) resolution. The results indicate that the range on vortex circulation RSME is [1.55-32.63] m^2/s, on vortex Y-position RSME is [0.73-134.51] and on vortex Z-position error is [1.187-10.11] m. The Y-position error had the largest variation with the maximum deviation exceeding 25m.

The second part sought to characterize the reduction in accuracy as the resolution of the input meteorological files was reduced from 1m to 100m. Normalized RMS values (normalized by the 1m solution) were calculated for resolutions of 5, 15, 20, 25, 30, 40, 100m over either the first 30 or first 60 seconds. For circulation the median for the resolutions greater than or equal to 30m was significantly different than the 5-25m resolutions at 30s, and at 60s, the deviation began at the 25m resolution. The median normalized RMS value for the 100m resolution was significantly different than the other resolutions for Y and Z position (30 & 60s). While the APA Suite predictions generally matched well with the measured vortex data, Y position had the largest variation. Similarly in the sensitivity study all of the resolutions for Circulation and Z position showed that the inter-quartile range remained within 5% of the 1m solution, that the inter-quartile range for the Y position at 100m exceeded 5% of the 1m solution.

As part of this effort the data provided were systematically transformed into a standard format required by the APA suite and evaluated (for the meteorological files) or a standard format for comparison with predicted files (for the wake vortex measurements). The resulting vortex prediction files and the measured wake vortex files were evaluated for defects and used in the sensitivity analysis. It was discovered that a dirth of information, both in terms of meteorological data and measured wake vortex data exists for altitudes above 1,400m.

TASK 2: WAKE VORTEX MODEL SCORING METHODOLOGIES

The objective of Task 2 is to develop consistent scoring methodologies to compare the performance of the both deterministic and probabilistic real-time wake vortex models. The task is split into two subtasks, as described at the beginning of this report.

Task 2.1 consists of a literature review on existing real time wake vortex prediction models and quantifies the assumptions within the models.

Task 2.2 establishes scoring methodologies for the wake vortex prediction models against test data. Presently, the RMS values of the differences between predicted and measured vortex strength, the differences between predicted and measured vortex (y,z) positions, and differences between time elapsed before the vortex leaves a predefined corridor are used to score models. Researchers also use 90 percentile errors to score the models. In this task, three scoring methodologies are established and automated via MATLAB codes. The purpose of these codes is to allow fast, consistent scoring of multiple wake vortex models against test data.

TASK 2.1: LITERATURE SURVEY

There has been considerable work done in the area of first-principles based modeling of aircraft wake. A variety of approaches have been proposed, that include Direct Numerical Simulations (DNS) on the behavior of vortices in turbulent shear flow, Large eddy simulations, Reynolds Averaged Navier-Stokes simulations that include a modeling of the wing (and airframe), Vortex particle methods and parabolized Navier-Stokes analyses. These approaches are very valuable for understanding the underlying physical phenomena, but are not presently suitable for real time vortex wake prediction.

Real-time vortex wake prediction methods need to be fast, and provide response in a fraction of a second, for implementation on aircraft. They should predict circulation and vortex position for a broad range of (wake generating) aircraft and atmospheric conditions. These need to be physics-based, and validated against high quality numerical simulations and exact solutions. A variety of approximate techniques have been developed over the past three decades, driven by the combined need for speed and accuracy.

<u>Representative Early Work 1920-1960</u>: For a detailed review of work done during 1920-1990, the reader is referred to Hallock's work which gives an annotated bibliography of aircraft wake vortex research [8]. The early models were validated using flight test data. Kraft has documented the velocity fields and persistence of trailing vortices from aircraft [9]. In this work, the trailing vortices were measured from a jet fighter airplane penetrating the wake of a propeller-driven small fighter airplane. The vertical and horizontal velocity components of vortices were measured using angle-of-attack vane and total pressure tube. It was noted that the vortices after 60 seconds (the longest time interval measured) from the shedding point still maintains large amount of circulation. Also noted was that the disturbance due to vortices was similar to severe turbulence and maintaining a precise course of an airplane flying in the trailing wake of another airplane was very difficult.

Because of the difficulties associated with measurement of wake velocity fields, researchers also used alternative approaches, e.g. looked at the load factors, elevator deflections to maintain 1g flight, and vertical displacement experienced by trailing aircraft [10]. Reeder and Wetmore [11] examined how the aircraft vortex wake affects terminal operations. Three modes of penetration of the trailing vortices were considered. These include the penetrating airplane flying at right angle to the trailing vortices, flying in a direction parallel to the vortices at midway of two vortex center lines (downwash region), and flying through vortex center line (where roll motion is expected due to upwash on one wing and downwash on the other wing). The effect of trailing vortices to three classes of airplanes including personal airplane, light transport and heavy transport were examined based on the equation for radial velocity distribution as a function of time, which accounts for vortex decay. Especially take-off and landing conditions at terminal area were considered, and suggestions were made for safe operations.

Robinson and Larson [12] examined four analytical methods for wake vortex modeling. Using four expressions of vortex-tangential velocity relation, computed rolling moments experienced by probe airplane were compared with measured flight test data. The four models were variation of tangential velocity expression related to initial circulation strength with different vortex decay model. The early three of the four models assumed flow viscosity (kinematic or eddy viscosity) plays dominant role in the decay rate. The fourth model, in conjunction with flight and wind-tunnel data, suggested that the circulation is proportional to the logarithm of the vortex core radius, and the decay rate of the tangential velocity is explicit function of downstream distance, while the core circulation is remained constant. None of the four models consistently correlated well with measured data. Instead, a simple interpolation between two models (two of vortex decay models based on flow viscosity) was suggested and it provided reasonably good prediction of minimum safe separation distance.

<u>Real Time Wake Vortex Models, 1970-2000</u>: Modern real time wake vortex models may be traced to the pioneering work of Crow [13] on the stability of a pair of trailing vortices. His work indicated that at sufficiently longer times, the trailing vortices will undergo a catastrophic demise event marked by a rapid growth in the core of the vortex and burst. He proposed a method to predict lifespan, the time at which vortex linking occurs, of aircraft trailing vortices as a function of dimensionless turbulence intensity, and the prediction was compared with measured data. Also suggested was a method to shorten lifespan of a wake by exciting mutual induction instability. Greene [14] developed a series of ODEs to model temporal evolution of vortices, impulse force, and vertical distance. The equation was obtained by balancing the time rate of change of impulse with external forces including ambient viscous force, buoyancy force due to density stratification, and atmospheric turbulence-generated viscous force. An empirical constant “q” was used to account for the effects of turbulence on vortex decay. A drag force term was used to model the interaction of the oval shaped vortex descending through air. Comparison with measured data from ground and flight test showed good agreement in vortex motion and decay, but further validation was required.

Kantha [15, 16] developed a semi-empirical approach that is similar to Greene’s method, and included ground proximity effects and cross wind shear. Using his model, several situations were considered to illustrate influences of ambient condition on the vortex decay. Vortex bounce near the ground was simulated, and accelerated vortex decay due to ambient turbulence and crosswind near the ground was simulated as well. Corjon and Poinsot [17] improved on Greene’s work, accounted for ground effects, and validated their work against Navier-Stokes and Idaho falls data.

The ground effect was modeled by image vortex and the constant vortex distance in the Greene's model was replaced to allow divergence of two vortices. Rebound of the vortices near the ground was simulated by including Liu's secondary vortex model [18] with modification, but without secondary image vortices. Crosswind and shear effect was modeled by including advection term, and the decay due to opposite sign vorticity from the crosswind was accounted in the sense of opposite circulation to the vortex. Finally, Crow instability, bursting, and core radius growth were accounted using simple empirical models. The model was validated against laminar Navier-Stokes solution and experimental data. It was shown that the dominant role of crosswind was advection of the vortices at out of ground. The rebound in simulation and experiment was explained by creation of secondary vortex. The rebound altitude was not constant, but related to core radius and Reynolds number.

Sarpkaya [19] postulated that the decay of the tip vortex prior to this burst event occurs in an exponential fashion with time. He replaced the quantity "q" with a parameter based on the eddy dissipation rate, ε. The drag force in Green's model was eliminated as physically inconsistent. The ground effect was modeled by a simple approach, which assumed constant decay rate ($d\Gamma/dt=d\Gamma/dt_{\_G}$) after the vortices enters ground effect region. In his new model, secondary vortices, which were introduced for vortex rebound modeling by early researchers, were not included and the only additional vortices introduced near the ground were the primary image vortices for zero-vertical velocity boundary condition. The model was compared with Memphis and DFW data, and showed reasonably good predictions. However, lack of the meteorological conditions of the field data gave difficulty in the model optimization.

Robins, Delisi, and Greene [20] extended Greene's model, and included ground effects, crosswind and vortex rebound. For ground effect, additional vortices were added sequentially as the trailing vortices enter from out of ground region to near ground, and in ground region. First, image vortices were included for zero-vertical velocity boundary condition at the ground as the vortices enter to near ground (image vortex region). As the vortices enter into "ground effect region", secondary vortices and their image vortices were introduced for rebound modeling. After the ground effect vortices have rotated 180 degree around primary vortices, second set of ground effect vortices were introduced with their corresponding image vortices. The vortex decay rate was hold constant after it entered image vortex region. There work was calibrated against the Memphis 1995 data. In a subsequent paper, they applied this model to Dallas data as well. Sarpkaya, Robins, and Delisi [6] subsequently combined Sarpkaya's 2000 model for the vortex decay with Robins et al's extensions for ground effects and rebound. The new model also includes new choice of vortex decay constant and inclusion of stratification effect on the circulation decay. The model was tested in DFW international airport and provided real time prediction of vortex evolution with favorable agreement with observed lidar data.

During the period, other approaches based on vortex particle methods were developed. A representative work is the approach by Zheng and Lim [21] where a vortex method is used to model interaction between trailing tip vortices and the shear layer. Ground rebound effects were included using image vortices. Vortex decay coefficient was determined from least square fit of measured circulation decay data, because measured TKE (as used in Greene's model) or turbulence eddy dissipation rate (as used in Sarpkaya's model) were not available from field data. The strength of shear vortices was held constant. The fast prediction from vortex method compared well with both NS solution with LES turbulence model and measured data. The asymmetry deflection of counter rotating two trailing vortices, that occurs when shear is located beneath or above the trailing vortices, was predicted reasonably well. Especially good agreement was found when the vortices are not too close to ground and the shear is strong. Sensitivity to wind measurement was also tested.

<u>Real Time Wake Vortex Models, 2000-Present</u>: The AVOSS prediction Algorithm (APA) has the following lineage. It started with Greene's model of how a vortex decays in atmosphere due to turbulence [14]. As discussed earlier, Greene's model was subsequently modified by Sarpkaya [6, 19] to eliminate the drag force term, and replace the turbulent kinetic energy related term with eddy dissipation rate term. As the trailing vortices evolve over time period after they are generated, the algorithm treats them through four phases. During the first phase, behavior of vortices is described by Sarpkaya's model until it feels ground. In phase II, as the vortices enter near ground (NGE region), the ground effects were modeled using image vortices [20]. Phase III begins when the vortices are close enough to the ground, where their interaction with ground produces secondary vortices. At this point, secondary vortices are added to the model. Phase IV is an extension of phase III and additional secondary vortices are generated. During phase II and later, the vortex decay rate is held constant at the rate of the end of phase I. A step-by-step implementation of the APA algorithm is described by Robins and Delisi [22]. Delisi and Robins [23] have compared the APA algorithm against measured data and 3-D LES simulations developed for a terminal area simulation system (TASS) [24] and show good correlations in the lateral transport of the trailing vortices with and without crosswind.

The APA algorithm and subsequent real time wake vortex models considerably benefited from the early LES simulations of Proctor [24, 25] where 3-D Navier-Stokes simulations with an explicit time marching algorithm, and subgrid models for Reynolds Stress were done. Proctor's work was originally intended for "the simulation of shallow cumulus to supercell cumulonimbus, including such convective phenomena as downbursts, tornadoes, gust fronts, and hailstorms." It was subsequently applied to "Terminal Area Simulation System (TASS)." In 1996, Switzer applied this algorithm to 3-D vortices [26], compared with exact solutions. In 1998-99, Proctor, and Han, used these LES simulations of a pair of vortices to extract empirical decay models for wake vortices as a function of time and ambient turbulent intensity. Proctor, Hamilton, and Switzer developed a simple set of ODEs based on the TASS LES work. This approach is called TASS Driven Algorithm for Wake Prediction (TDAWP) [27]. In this algorithm, the circulation strengths for vortex descent and vortex hazard were computed separately, whereas previous vortex prediction methods (e.g. Greene, Sarpkaya, or Holzäpfel's model) used single equation for both. The effect of Ground and nonlinear crosswind shear were not included.

<u>Probalistic Two-Phase Wake Vortex Decay and Transport Model (P2P)</u>: This model was developed by Holzäpfel [28]. It accounts for the stochastic nature of turbulence, vortex instabilities, and deformations using a probabilistic variation. The uncertainties and fluctuations in environmental and aircraft parameters are also captured using bounds. The vortex Evolution is described by a diffusion phase, followed by rapid decay phase. Results from this model are presented as confidence intervals for vortex strength and position. In a subsequent work [29], this model was enhanced as follows. The vortex age and descent speed are adjusted to match effects of axial wind and glide-slope angle. The probabilistic envelopes were expanded to consider tilting, stalling, and rebounding of wake vortices caused by axial- and crosswind shear. Detailed comparisons with the

APA model were made by Holzäpfel in 2006. Procter and Hamilton have compared APA, TDAWP, and D2P [4].

TASK 2.2: DEVELOPMENT OF SCORING METHODOLOGIES

In order to rate different real-time wake vortex models, a consistent set of scoring methodologies was developed to evaluate model performance in a multitude of regimes. In previous work by Robins and Delisi [30], wake vortex model scoring methodologies consisted of the RMS of the error between the measured and predicted vortex strength and position, 90th percentile of the error, and the measured exit time vs. predicted exit time of the vortex from a predefined corridor. The latter method was originally developed by Hinton [31] for the Aircraft Vortex Spacing System (AVOSS).

In this section, three methodologies are presented which accomplish these goals. A normalized residual method handles the low-level accuracy of the wake vortex models, while high-level accuracy is handled by the corridor exit method defined by Hinton. The last method scores mid-level accuracy using a combination of the accuracy of the data and the accuracy of the behavior of the data in the form of a correlation coefficient. This allows scoring of a particular dataset to be tailored to what is required by the user. For example, a pilot may only care about the high-level scoring, whereas a real time wake vortex model developer may be more interested in the low-level accuracy of different models.

METHOD 1: RESIDUALS OF TIME SERIES

The simplest accuracy scoring method is to plot both the measured data and predicted data alongside one another vs. time. This allows for a qualitative assessment of the accuracy of a particular vortex prediction model over the entire prediction horizon, but provides few quantitative metrics for comparison. This method was previously used by Robins and Delisi [30]. By considering the normalized residuals of the data, a clearer picture of the accuracy can be obtained:

$$\overline{x}_{res} = \frac{x_{meas} - x_{pred}}{x_0} \tag{3}$$

Here, x_{meas} is the measured data (vortex circulation strength, vortex x-position, or vortex y-position), x_{pred} is the corresponding predicted data, x_0 is the normalization factor (initial circulation strength or aircraft wingspan, depending on the data type), and $\overline{x}_{res}$ is the resulting non-dimensional residual. Using this method, the residuals can be compared across multiple aircraft types and conditions while minimizing ambiguity due to aircraft size or vortex strength. The mean residual can be calculated for different phases of the vortex to develop an overall picture of the accuracy of the model during the entire vortex evolution.

While this method in itself can be useful for evaluating small datasets, due to the large number of datasets available in Task 1, the normalized residuals are averaged over a subset of the datasets at each time step. For example, if the normalized residuals for the APA model at MEM airport for large aircraft are being calculated, they are averaged over all the APA/MEM/large cases so that an overall trend can be observed.

METHOD 2: CORRIDOR EXIT TIME

To ascertain the high level accuracy of each method, the corridor exit time developed by Hinton [31] is used. This method compares the exit time of the measured vortex and the predicted vortex from a predefined flight corridor. If the measured vortex leaves the corridor before the predicted corridor—that is, the measured exit time is smaller than the predicted exit time—then it can be said that the model provides a conservative prediction of the vortex exit. For a pilot, this means that the actual vortex can be all but guaranteed to have left the corridor when the model predicts that it has. The opposite case is that where the measured vortex leaves the corridor *after* the predicted vortex. This is not desirable because the model would provide a false positive for vortex exit to the pilot.

The method is detailed as follows: A corridor in 3D space is defined behind a generated vortex. This can represent the area through which a following aircraft will pass after a given time. The time for the measured vortex to exit this defined corridor is plotted against the time for the predicted vortex to do the same. According to Hinton, a typical aircraft landing corridor is defined as 1000 ft wide by 200 ft tall at initial approach. In the MATLAB codes, this corridor size is user adjustable.

For the y-position, the exit time is defined as:

$$\left|Y(t_{ey})\right| - \tfrac{1}{2}C_W = 0 \tag{4}$$

where Y is the vortex y-position, t_{ey} is the exit time for the y-coordinate of the vortex, and C_W is the corridor width. A similar equation is defined for the z-coordinate:

$$\left|Z(t_{ez}) - Z(0)\right| - \tfrac{1}{2}C_H = 0 \tag{5}$$

Here, Z is the vortex z-position, t_{ez} is the exit time for the z-coordinate of the vortex from the corridor, and C_H is the corridor height. Since the z-position is measured relative to the ground, the initial z-position must be subtracted to obtain the distance from the center of the corridor.

The vortex exit time, t_{exit} is defined as the lesser of t_{ey} and t_{ez}:

$$t_{exit} = \min(t_{ey}, t_{ez}) \tag{6}$$

If the above expression produces an exit time equal to the final time (i.e. the vortex never exited the corridor for the data available or died within the corridor), the vortex strength is checked again to determine if the vortex died within the corridor. If this occurred, the corresponding time is recorded as the corridor exit time. If the exit time calculated is equal to the final time and no vortex death occurred within the corridor, the corridor exit time is undefined.

Ideally, if the vortex prediction method is accurate, both the predicted and measured exit times will be equivalent, and the data on a plot of one versus the other will lie in a line of unity slope with its origin at zero. If either the measured or predicted vortex does not leave the corridor, the exit time is not defined and therefore is not plotted.

METHOD 3: CROSS-CORRELATION

When comparing two sets of time series data, a good measure to determine how well they compare is the cross-correlation. This method has been previously used to uniquely identify vortices as they progress in time [32]. It provides a mid-level measure of accuracy when comparing measured and predicted data by taking into account both accuracy and behavior of the two data sets. Here, it is used to measure the uncertainty between the measured and predicted data, as well as the uncertainty between different resolutions of predicted data. Specifically, it assists in determining if the measured data and predicted data are properly aligned in time. If they are not, the peak correlation will occur at a lag time not equal to zero (lag time is defined below, after the cross-correlation equation).

When the lag time is set to zero, the correlation between two sets of data can be readily calculated. This value scores how close the behavior of the two sets of data is as well as how close two sets of data match each other. This allows one to distinguish between cases with similar residuals but different behavior. Ideally, residuals would be low and correlation would be high between two sets of data. If residuals rise in value but the correlation also remains high, this shows that while the two data sets do not match exactly, they do behave in the same manner. Thus it can be assumed that the underlying processes responsible for each data set are similar.

The normalized cross-correlation function is defined here:

$$R_{xy}(l) = \frac{\sum_{n=0}^{N-l} x(n+l)\,y(n)}{\sqrt{\sum_{n=0}^{N} x^2(n)}\sqrt{\sum_{n=0}^{N} y^2(n)}} \tag{7}$$

Here, x and y represent vortex circulation strength, x-position, or y-position of two different data sets. N is the final time, l is the lag time, and n is the current time. The lag time is used to measure how long two data sets remain well correlated with each other. The cross-correlation is normalized by the product of norms of x and y to allow for easier comparison among several cases. Values close to unity would indicate well correlated data. As l increases, x and y will become less correlated, so $R_{xy}(l)$ should go to zero.

The correlation calculation can compare experimental data vs. predicted data, different methods (APA vs. TDP), or smaller time periods within the full time series to determine correlation at particular vortex phases.

SUMMARY

Three methodologies have been introduced to score wake vortex models against test data: a residual method, a corridor exit method, and a cross-correlation method. Each method is useful for measuring different levels of accuracy: low-level, high-level, and mid-level, respectively. Ideally, to score a particular wake vortex model, one would choose the method that is most represents the level of accuracy required. For example, the high-level accuracy scoring method may be most useful for pilots, who simply want to know when a vortex is guaranteed to have left a prescribed corridor.

Mid-level and low-level accuracy may be more useful for researchers who are attempting to improve the resolution of wake vortex models.

In the next section, these scoring methodologies will be used to score the APA and TDP wake vortex models from the APA Suite against provided test data from Task 1.

TASK 3: SCORING RESULTS OF AVAILABLE MODELS

In this task, the methodologies, tools, and rationales developed in Task 2 are used to score the wake vortex models against experimental data identified in Task 1. The wake models used are the APA and TDP vortex prediction models from the APA Suite. The sensitivity of the accuracy of these models was previously analyzed in Task 1. Here, different measures of accuracy are applied to the models to determine if they differ in different regimes of prediction.

At each site, the results are grouped by aircraft size and model type. In the past, only a limited set of data (Memphis, Dallas Fort Worth, and Denver) has been used using a limited number of models. The current study performs a more comprehensive analysis using all the models available (APA and TDP) to the researchers, and all the data that has been certified under Task 1, using the scoring methodologies developed under Task 2.

MATLAB codes of the developed methodologies are presented in the Appendix along with basic usage instructions.

NORMALIZED RESIDUALS RESULTS

Scoring using the residual method was done to determine if a significant difference could be seen between the APA and TDP vortex prediction methodologies in the APA Suite. Data presented in Task 1 was used to complete this evaluation. Plots of the residuals are presented for small, large, and heavy aircraft at DFW and MEM for both prediction models. The data is plotted with respect to the distance of an aircraft measured from the lead aircraft, assuming that the following aircraft is traveling at 180 kts relative to the deposited vortex, a typical jet aircraft initial approach speed. This is simply a rescaling of the original time scale to provide some insight into how time translates to a distance. This speed is user adjustable in the included MATLAB code.

Figure 44 shows the normalized circulation residuals for small aircraft at DFW and MEM airports. The mean residual at each following distance is plotted with +/- three standard deviations (3σ) to get an overall picture of where the majority of the data lies. There was very limited data available for small aircraft, so general conclusions cannot be extended from these results. The +/- 3σ lines on the plot change sporadically because the same number of measured datasets is not available for all following distances. For larger following distances, data is more scarce, so the standard deviation shrinks dramatically.

Figure 45 and Figure 46 show the Y- and Z-position residuals for small aircraft at DFW and MEM airports. The range of the data gradually rises from zero wingspans to 8-10 wingspans for the Y-position and 2-4 wingspans for the Z-position as the following distance increases to approximately 1200 m. This illustrates that when the following aircraft is close to the vortex generating aircraft, the prediction has much smaller variance than at larger distances.

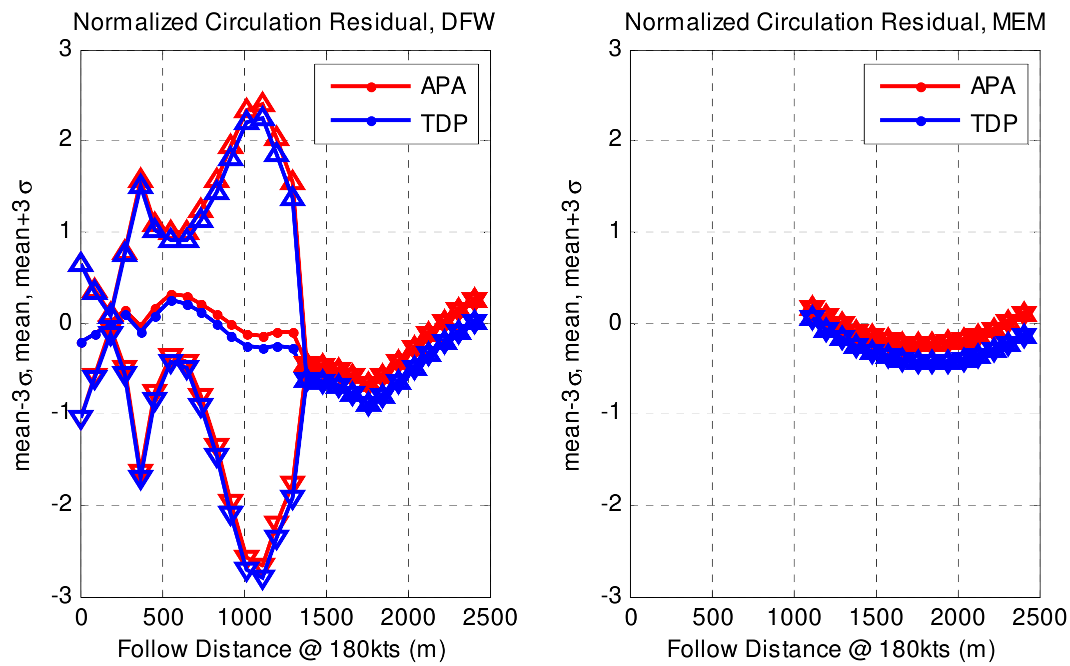


FIGURE 44. NORMALIZED CIRCULATION RESIDUALS FOR **SMALL** AIRCRAFT AT DFW AND MEM. MEAN AND STANDARD DEVIATION IS TAKEN AT EACH FOLLOW DISTANCE.

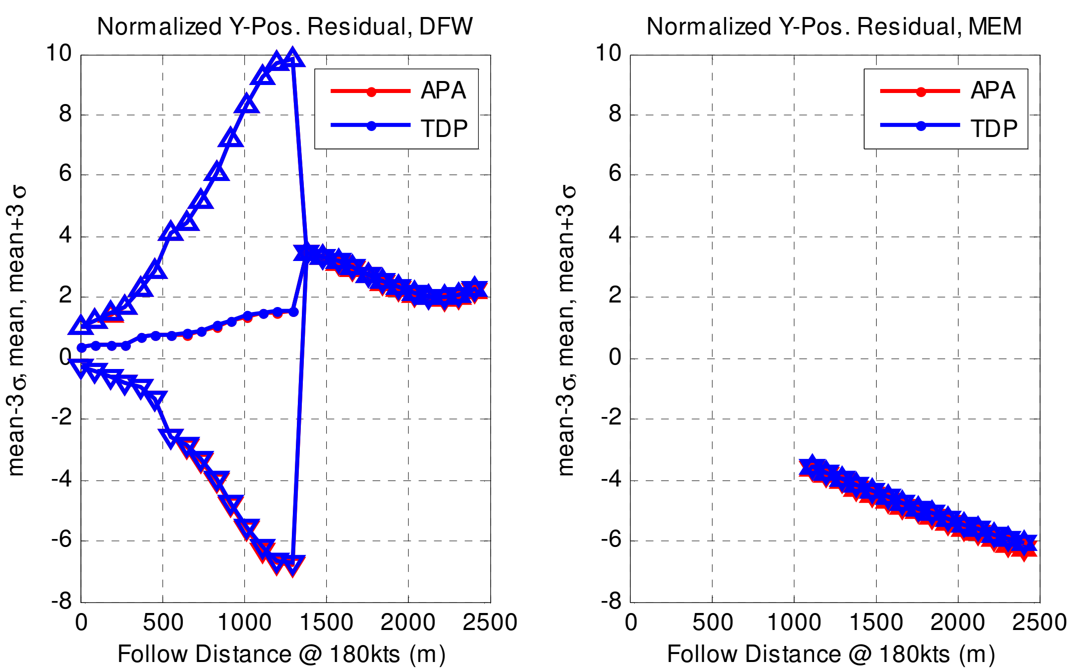


FIGURE 45. NORMALIZED Y-POSITION RESIDUALS FOR **SMALL** AIRCRAFT AT DFW AND MEM. MEAN AND STANDARD DEVIATION IS TAKEN AT EACH FOLLOW DISTANCE.

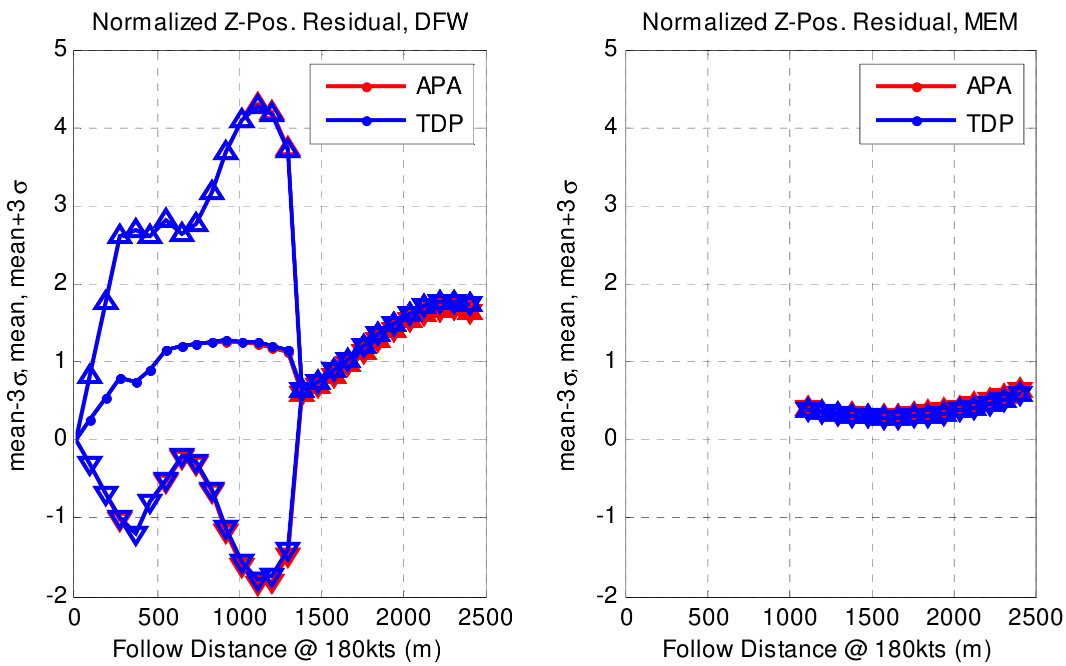


FIGURE 46. NORMALIZED Z-POSITION RESIDUALS FOR **SMALL** AIRCRAFT AT DFW AND MEM. MEAN AND STANDARD DEVIATION IS TAKEN AT EACH FOLLOW DISTANCE.

Figure 47 shows the circulation residuals for large aircraft at DFW and MEM. Compared with the small aircraft, the variance appears to be approximately double, but this could be just due to the larger number of datasets available for large aircraft. As before, MEM shows much smaller variance in the data as DFW, which is indicative of the difference in measurement methods used at the two locations. There is no clear difference between the two prediction models in terms of scoring. This confirms the observations from Task 1, since the residual method is a measure of accuracy. There is very little difference in the accuracy of each method.

Figure 48 and Figure 49 show the corresponding Y-position and Z-position residuals, respectively. Here, again, the Y-position residual shows significantly larger variance than the Z-position. This confirms the similar results shown in Task 1: Y-position prediction appears to be more uncertain across the dataset than Z-position. Between locations, the variance in Y-position is still slightly smaller at MEM than at DFW, but the disparity is reduced compared to that for the circulation residual. The Z-position residual variance is comparable at both locations.

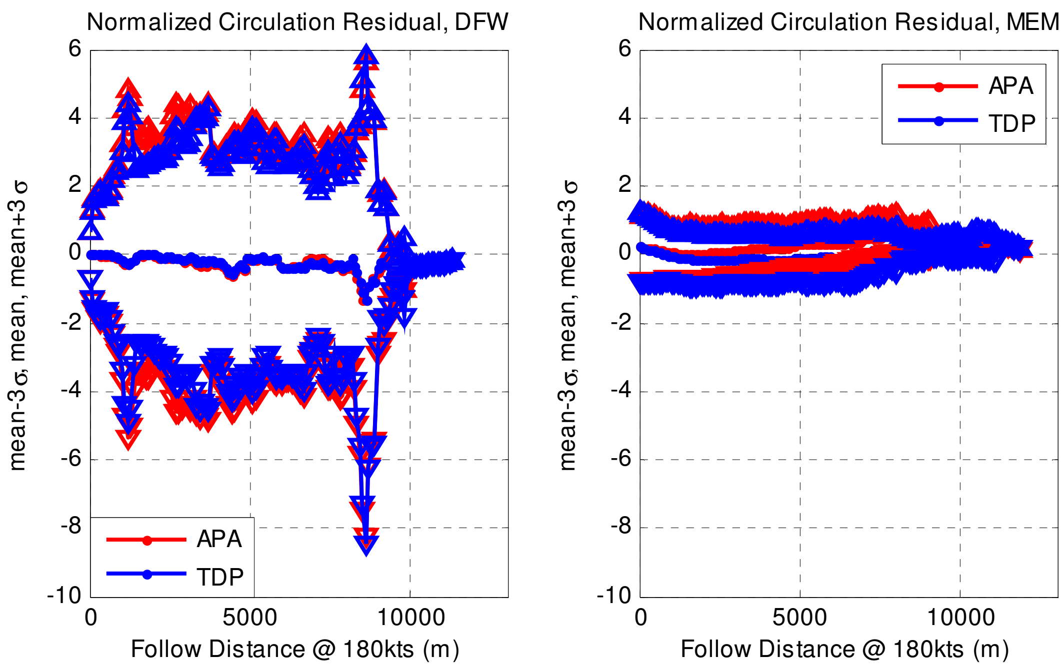


FIGURE 47. NORMALIZED CIRCULATION RESIDUALS FOR **LARGE** AIRCRAFT AT DFW AND MEM. MEAN AND STANDARD DEVIATION IS TAKEN AT EACH FOLLOW DISTANCE.

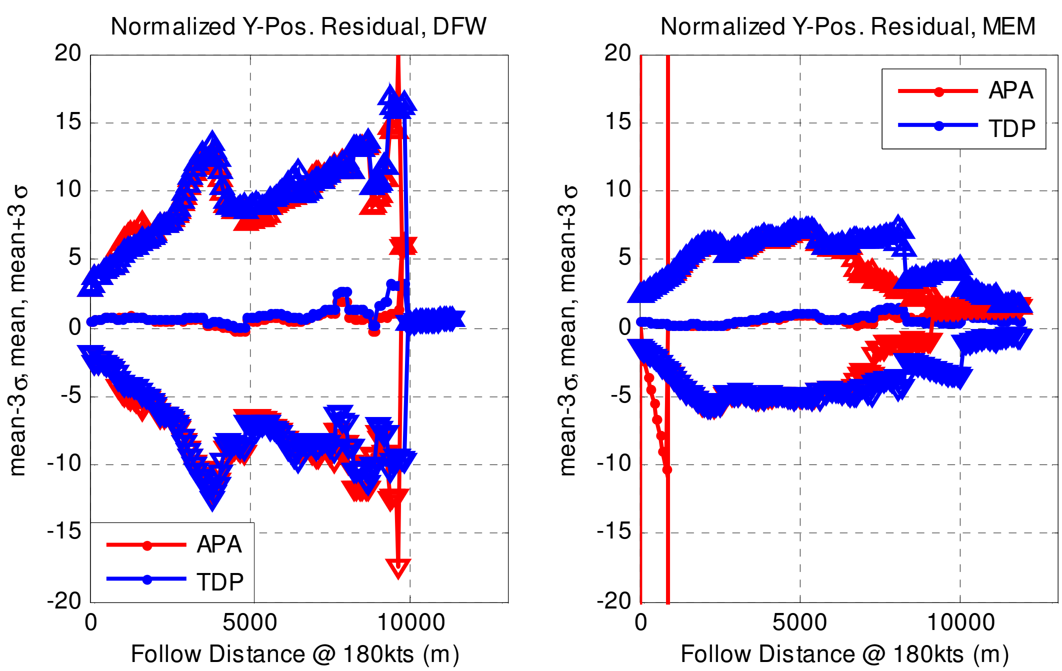


FIGURE 48. NORMALIZED Y-POSITION RESIDUALS FOR **LARGE** AIRCRAFT AT DFW AND MEM. MEAN AND STANDARD DEVIATION IS TAKEN AT EACH FOLLOW DISTANCE.

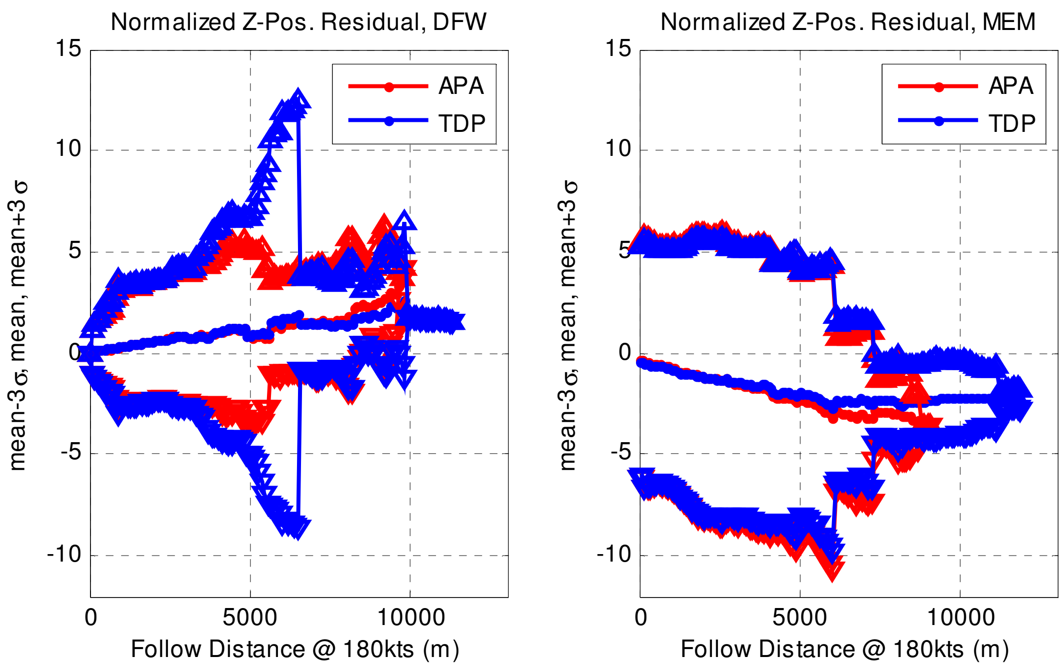


FIGURE 49. NORMALIZED Z-POSITION RESIDUALS FOR **LARGE** AIRCRAFT AT DFW AND MEM. MEAN AND STANDARD DEVIATION IS TAKEN AT EACH FOLLOW DISTANCE.

Figure 50 shows the circulation residuals for heavy aircraft at DFW and MEM airports. Here, we see a similar result to the large aircraft case: the circulation residual variance is comparable and the diminished nature of the variance at MEM is also comparable. There is no clear difference between the two models. Figure 51 and Figure 52 show the results for Y-position and Z-position residuals. The confidence interval of the Y-position residual at MEM is about half the size at DFW. This can be attributed to the difference in measurement equipment at each site. The Z-position confidence interval is larger in MEM than at DFW. This follows the trend seen with large aircraft as well, and can again be attributed to a difference in measurement equipment.

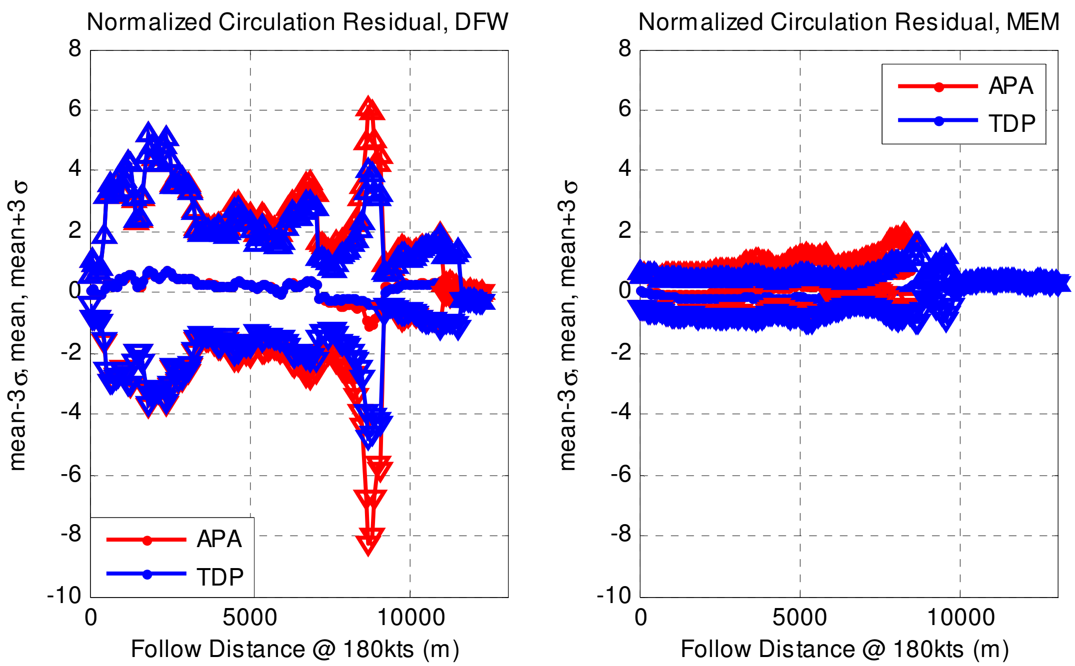


FIGURE 50. NORMALIZED CIRCULATION RESIDUALS FOR **HEAVY** AIRCRAFT AT DFW AND MEM. MEAN AND STANDARD DEVIATION IS TAKEN AT EACH FOLLOW DISTANCE.

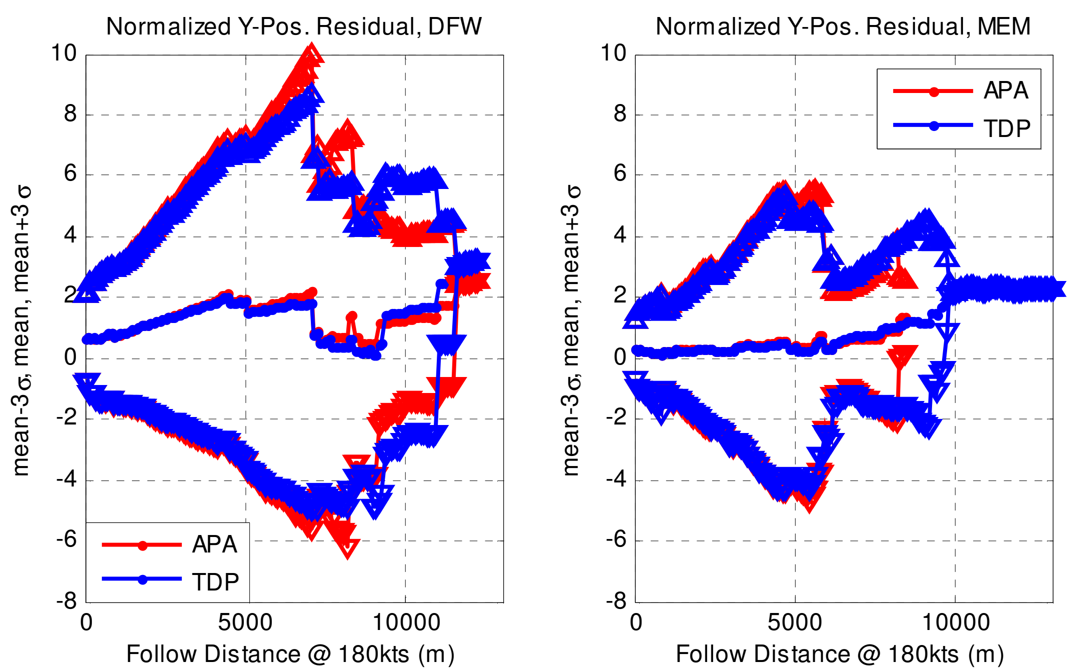


FIGURE 51. NORMALIZED Y-POSITION RESIDUALS FOR **HEAVY** AIRCRAFT AT DFW AND MEM. MEAN AND STANDARD DEVIATION IS TAKEN AT EACH FOLLOW DISTANCE.

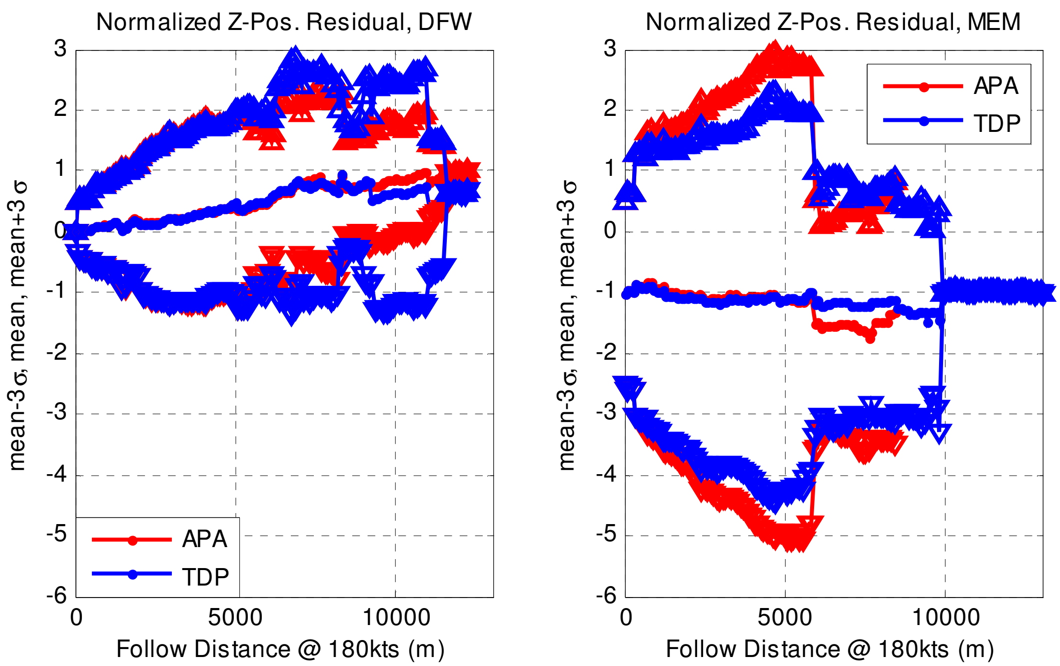


FIGURE 52. NORMALIZED Z-POSITION RESIDUALS FOR **HEAVY** AIRCRAFT AT DFW AND MEM. MEAN AND STANDARD DEVIATION IS TAKEN AT EACH FOLLOW DISTANCE.

Overall, the residual results show that the measurement methods in MEM are superior to those in DFW for circulation and Y-position, but seem to be inferior for Z-position. Table 19 and Table 20 show the largest mean residuals for the APA and TDP models, respectively, at both locations. For large and heavy aircraft, circulation and Y-position have smaller maximum mean residuals at MEM, further illustrating this point. The Z-position results are less conclusive, but from the plots above, it is clear that the variance is lower at the MEM location.

TABLE 19. LARGEST MEAN RESIDUALS – APA MODEL.

	DFW			MEM		
	Circ.	**Y-Pos**	**Z-Pos**	**Circ.**	**Y-Pos**	**Z-Pos**
Small	-0.67	3.48	1.67	-0.23	-6.27	0.64
Large	-1.34	3.38	3.05	0.59	1.67	-3.54
Heavy	-1.09	2.16	0.93	-0.19	1.36	-1.75

TABLE 20. LARGEST MEAN RESIDUALS – TDP MODEL.

	DFW			MEM		
	Circ.	**Y-Pos**	**Z-Pos**	**Circ.**	**Y-Pos**	**Z-Pos**
Small	-0.87	3.48	1.78	-0.43	-6.09	0.59
Large	1.33	3.34	2.37	0.28	1.78	-2.56
Heavy	-0.54	2.04	0.96	0.67	1.16	-1.23

CORRIDOR EXIT TIME RESULTS

The previously described corridor exit method was evaluated using the available data from the APA and TDP models to determine if the models showed a significant difference in the ability to predict vortex corridor exit times. Hinton's corridor definition of 1000 ft width x 200 ft height was used, but other definitions for vortex corridors may also be used via user specification in the MATLAB code. The 1000 ft x 200 ft corridor corresponds to a position 5 nm from the runway [31]. The corridor exit times for the measured and predicted data were calculated as shown above, and then plotted for small, large, and heavy aircraft at the two locations available, DFW and MEM. The data used to generate these results are the same as those presented in Task 1.

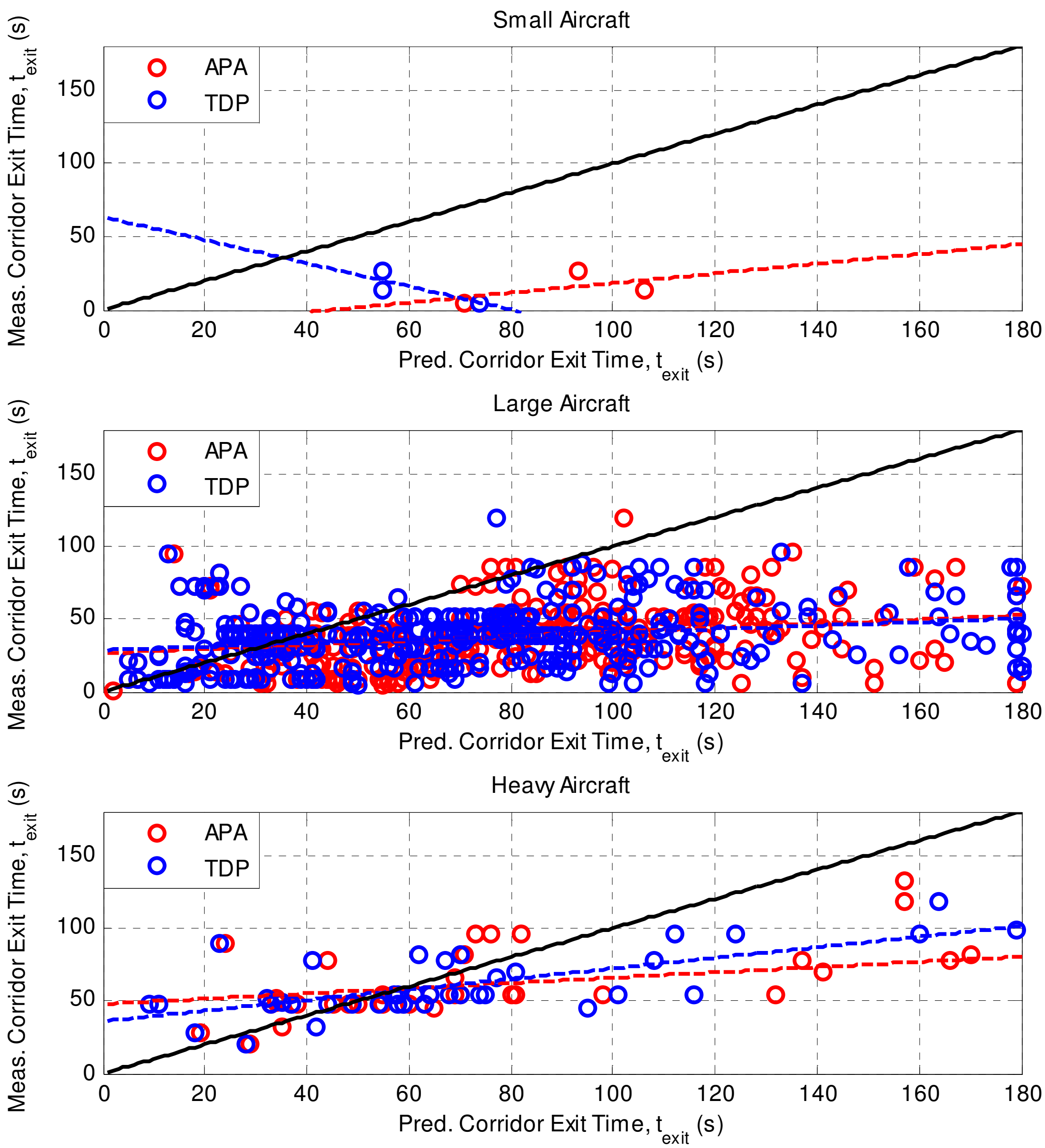


FIGURE 53. MEASURED CORRIDOR EXIT TIME VS. PREDICTED CORRIDOR EXIT TIME FOR SMALL, LARGE, AND HEAVY AIRCRAFT AT **DFW** AIRPORT; DASHED LINES ARE LEAST SQUARES FIT. **1000FT X 200FT** CORRIDOR.

Figure 53 shows the measured corridor exit time vs. the predicted corridor exit time for aircraft at DFW airport. There were very limited cases available for small aircraft, so little can be said other than that both methods over-predict the time to leave the corridor. For the large and heavy aircraft, enough data points were available to warrant a linear least squares fit, shown in the dotted lines. The majority of the points lie below the black line in these cases, indicating that again, the methods tend to over-predict the corridor exit time. As stated in the Task 2 description of this method, this is a more conservative condition, and is therefore better than under-predicting the exit time.

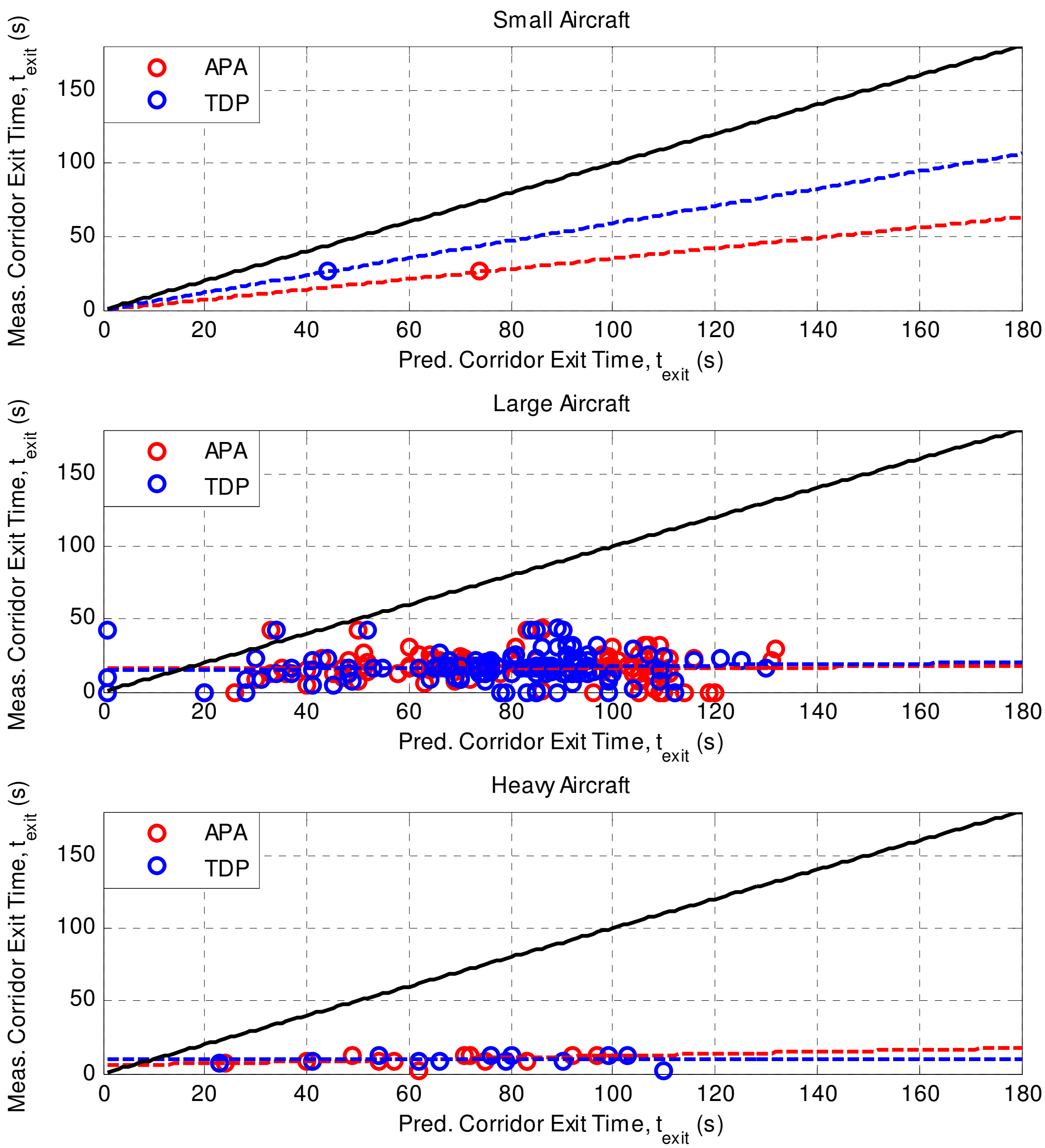


FIGURE 54. MEASURED CORRIDOR EXIT TIME VS. PREDICTED CORRIDOR EXIT TIME FOR SMALL, LARGE, AND HEAVY AIRCRAFT AT **MEM** AIRPORT; DASHED LINES ARE LEAST SQUARES FIT. **1000FT X 200FT** CORRIDOR.

Figure 54 shows the same plot for the MEM airport cases. At this location, similar trends are observed, but there is much more variation in the results. For approximately the same measured vortex exit times, the predicted exit times vary through a large range (thus the slope of the resulting line of best fir is nearly zero). However, for the majority of the cases, both the APA and TDP methods still provide a conservative estimate of the exit time. If the predicted exit time is above ~40 seconds, the models consistently provide a conservative estimate of the exit time. Below ~40 seconds, the models may under-predicted the corridor exit time, which is not desired. Judging by the slope of least squares fit, it appears that TDP methodology performs slightly better than the APA methodology for large and heavy aircraft.

Table 21 shows the slopes and intercepts associated with the lines of best fit for each case in Figure 53 and Figure 54. Since there were very few small aircraft cases, these slopes and intercepts can be ignored as they do not provide a reasonable measure of the population. In all cases except for heavy aircraft at MEM using the TDP method, it appears that the heavy aircraft produce a slope closer to unity, indicating that corridor exit times are more accurately predicted for heavier aircraft. The lower slope for the heavy aircraft at MEM using TDP is not statistically significant because of the small number of points associated with that category.

TABLE 21. SLOPES AND INTERCEPTS OF LINE OF BEST FITS FOR CORRIDOR EXIT METHOD WITH **1000FT X 200FT** CORRIDOR.

		DFW		MEM	
		Slope	**Intercept**	**Slope**	**Intercept**
APA	**Small**	0.3307	-14.7604	0.3514	0
	Large	0.1437	26.6520	0.0008	16.8407
	Heavy	0.1804	47.8024	0.0659	5.6543
TDP	**Small**	-0.7895	63.4211	0.5909	0
	Large	0.1208	28.5746	0.0315	14.4487
	Heavy	0.3628	35.9206	0.0031	9.6915

Results for another corridor definition are presented as well. This time, the corridor height is 537 ft and the width is 114 ft. This corresponds to a distance of 2 nm from the runway [31]. Similar plots for DFW and MEM are shown in Figure 55 and Figure 56, respectively. It appears that the results are similar for both locations, except for heavy aircraft at DFW airport. Here, the majority of the points lie in the “under-prediction” area of the plot (above the black line marking unity slope). The same is not true for heavy aircraft at MEM airport. At the MEM location, the predicted corridor exit time to guarantee a conservative estimate of corridor exit is smaller than that of the previous corridor definition. It is now roughly half, from ~40 seconds to ~20 seconds. Thus, for smaller corridor sizes, a shorter simulation is required before it can be guaranteed that the real vortex has left the corridor.

Table 22 shows the slopes and intercepts of the lines of best fit for the 537 ft x 114 ft corridor exit plots. Again, due to the limited number of small aircraft cases, nothing can be gleaned from the line of best fit in these cases. However, the same trend is evident as before for the large and heavy aircraft—as weight increases, the slope also increases, meaning the corridor exit time is more accurately predicted.

TABLE 22. SLOPES AND INTERCEPTS OF LINE OF BEST FITS FOR CORRIDOR EXIT METHOD WITH **537FT X 114FT** CORRIDOR.

		DFW		MEM	
		Slope	**Intercept**	**Slope**	**Intercept**
APA	**Small**	0.3750	-8.8750	1.0833	0
	Large	0.0951	24.6395	0.0435	9.4349
	Heavy	0.1802	41.4448	0.2678	2.6175
TDP	**Small**	1.1053	-34.7895	1	0

	Large	0.1138	23.6319	0.0418	9.0289
	Heavy	0.1285	42.9242	0.2379	3.0044

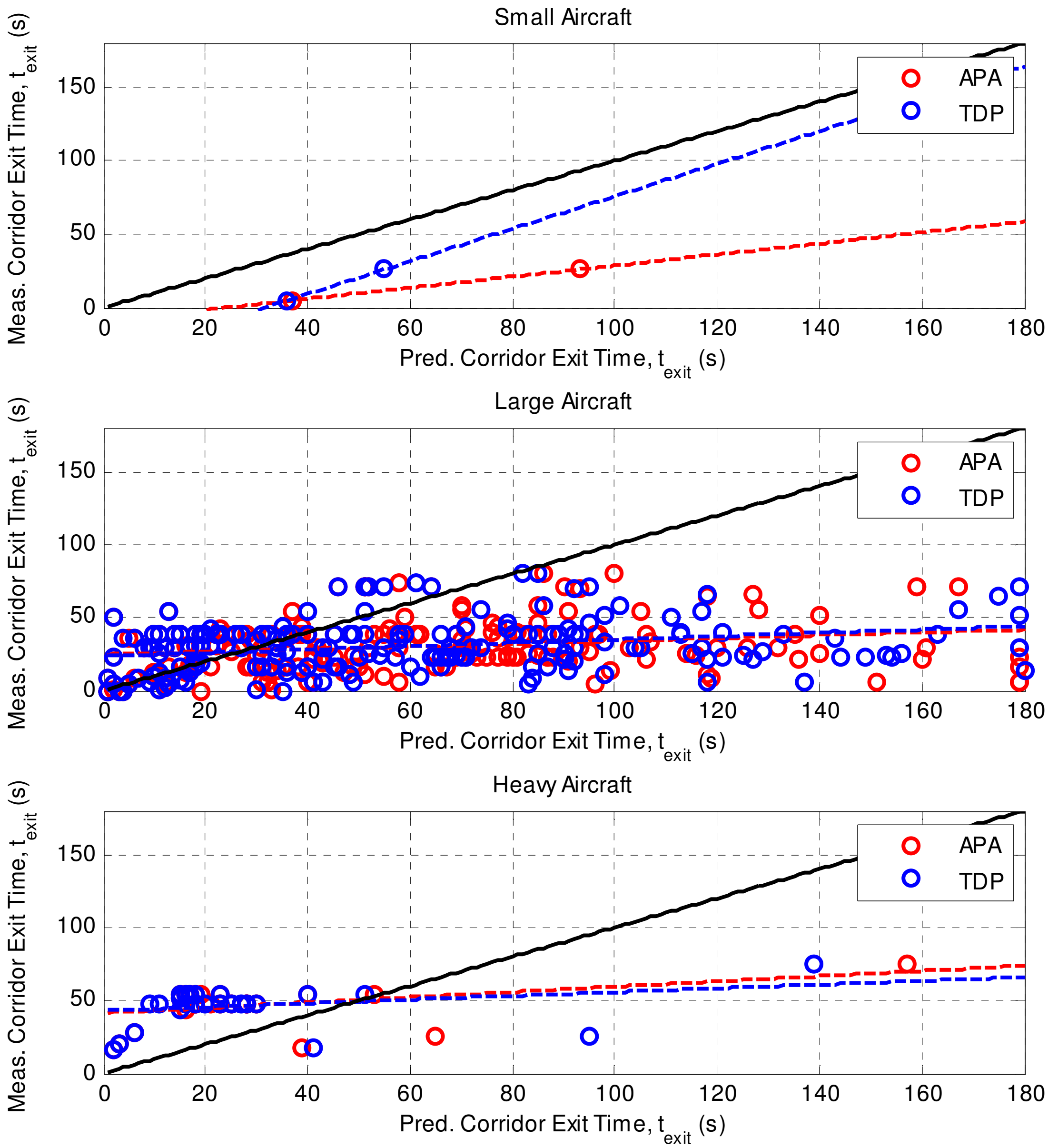


FIGURE 55. MEASURED CORRIDOR EXIT TIME VS. PREDICTED CORRIDOR EXIT TIME FOR SMALL, LARGE, AND HEAVY AIRCRAFT AT **DFW** AIRPORT; DASHED LINES ARE LEAST SQUARES FIT. **537FT X 114FT** CORRIDOR.

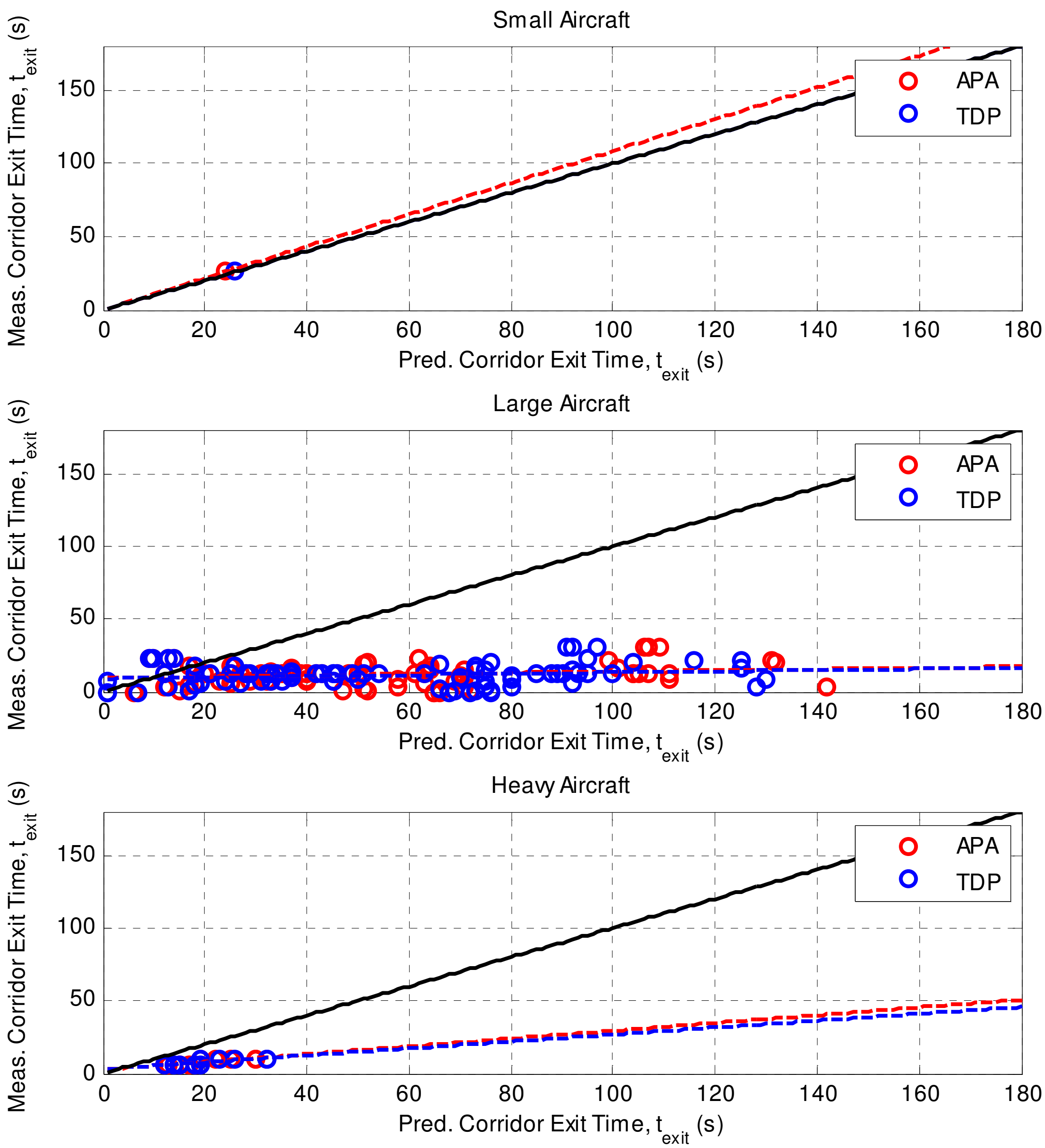


FIGURE 56. MEASURED CORRIDOR EXIT TIME VS. PREDICTED CORRIDOR EXIT TIME FOR SMALL, LARGE, AND HEAVY AIRCRAFT AT **MEM** AIRPORT; DASHED LINES ARE LEAST SQUARES FIT. **537FT X 114FT** CORRIDOR.

CROSS-CORRELATION RESULTS

An evaluation of the cross-correlation method was done to determine how well it differentiated the two analysis methods (APA and TDP). The cross-correlation coefficients were calculated for circulation, Y-position, and Z-position. The first cross-correlation was calculated with the predicted data at 1m height resolution (x_1) and the measured data (x_m) in order to assess the accuracy of each method. The second cross-correlation was calculated with x_1 and the predicted data at 100m height resolution (x_{100}) to assess the effect of height resolution on the predictions.

The results are then plotted against lag time for large and heavy aircraft at the two locations available, DFW and MEM. The small aircraft case was neglected because of the miniscule amount of cases available for comparison. The data used to generate these results are the same as those presented in Task 1.

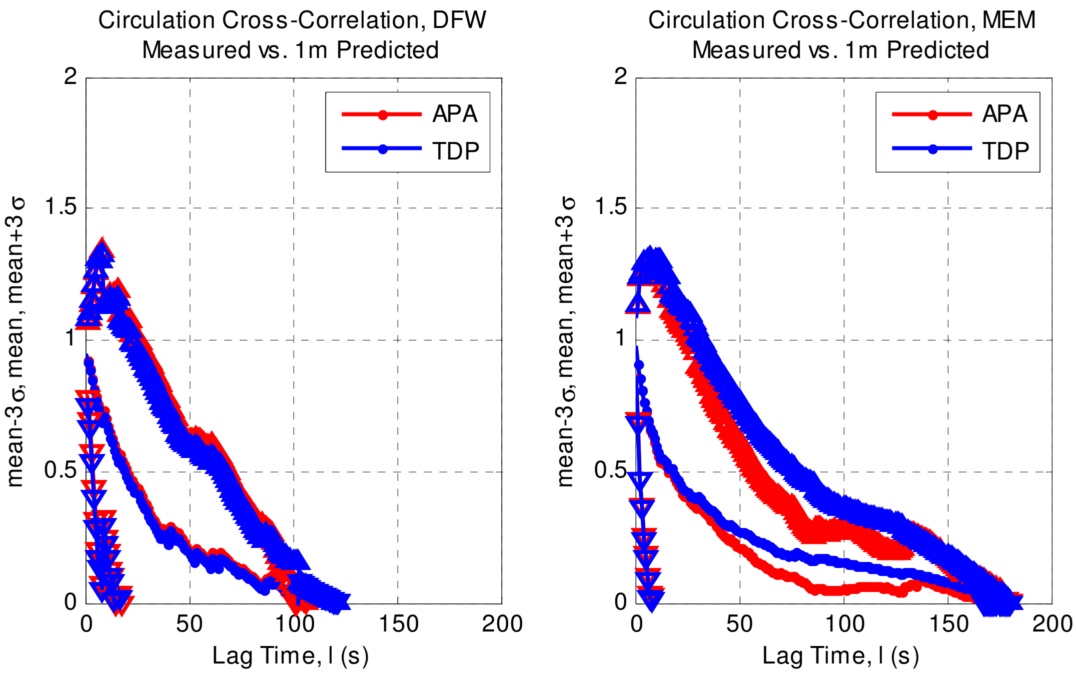


FIGURE 57. MEAN CIRCULATION CROSS-CORRELATION OF MEASURED AND 1M RESOLUTION PREDICTED DATA FOR **LARGE** AIRCRAFT. UPPER AND LOWER BOUNDS (TRIANGLES) SHOW +/- 3 STANDARD DEVIATIONS FROM MEAN (POINTS).

Figure 57 shows the mean circulation cross-correlation between measured and 1m resolution predicted data for large aircraft at both DFW and MEM. The mean is computed at each lag time over all available data sets. Also plotted are the +/- 3 standard deviation curves from the mean, to indicate the range of the data. From this plot, there is very little difference between the two models in terms of correlation at DFW, but at MEM, TDP maintains higher correlation at higher lag times than APA. This means that the circulation changes less dramatically from the initial measured data as time forward in the TDP model. Depending on how dramatically the measured data changes, this quality can be favorable or unfavorable.

Figure 58 shows the same plot, but for Y-position cross-correlation. Here the same phenomenon is observed at the MEM location; the TDP model maintains a higher correlation at higher lag times than the APA model. Figure 59 shows the Z-position cross-correlation with similar results. The jagged nature of the mean in the DFW location simply means that not many data sets extended to that time period, and thus the diminished number of datasets causes a sudden change in mean. This is not indicative of a physical phenomenon.

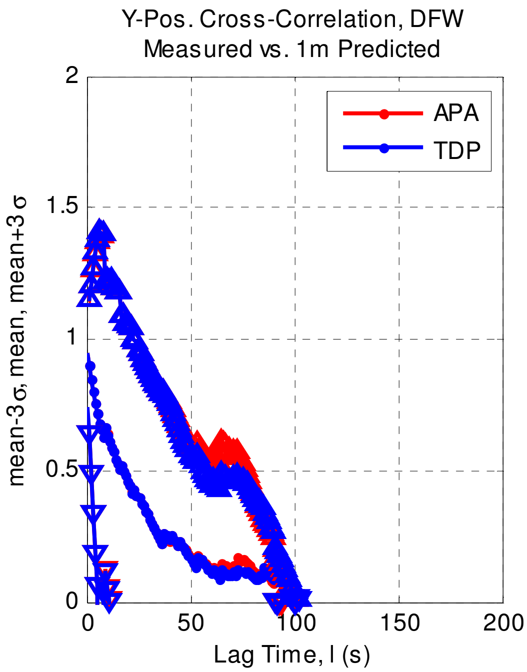


FIGURE 58. MEAN Y-POSITION CROSS-CORRELATION OF MEASURED AND 1M RESOLUTION PREDICTED DATA FOR **LARGE** AIRCRAFT. UPPER AND LOWER BOUNDS (TRIANGLES) SHOW +/- 3 STANDARD DEVIATIONS FROM MEAN (POINTS).

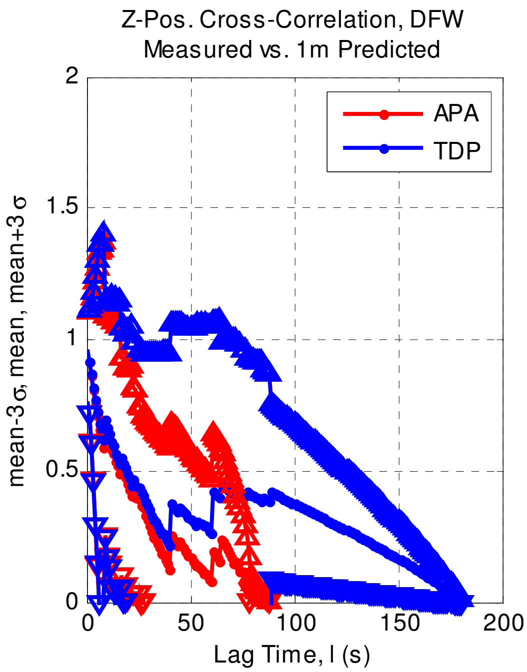


FIGURE 59. MEAN Z-POSITION CROSS-CORRELATION OF MEASURED AND 1M RESOLUTION PREDICTED DATA FOR **LARGE** AIRCRAFT. UPPER AND LOWER BOUNDS (TRIANGLES) SHOW +/- 3 STANDARD DEVIATIONS FROM MEAN (POINTS).

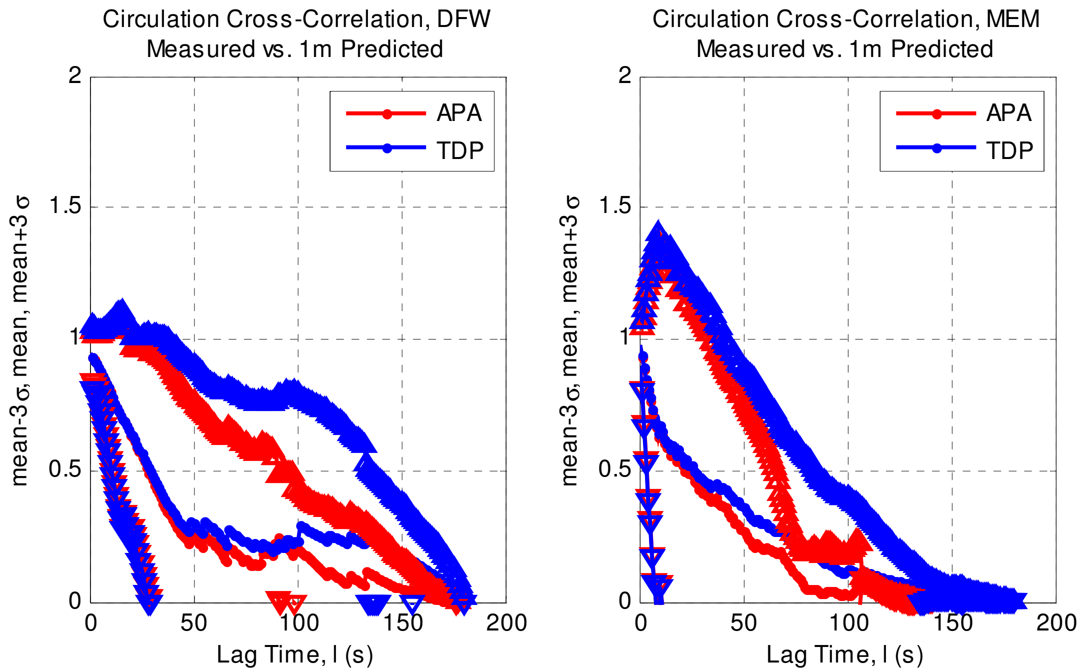


FIGURE 60. MEAN CIRCULATION CROSS-CORRELATION OF MEASURED AND 1M RESOLUTION PREDICTED DATA FOR **HEAVY** AIRCRAFT. UPPER AND LOWER BOUNDS (TRIANGLES) SHOW +/- 3 STANDARD DEVIATIONS FROM MEAN (POINTS).

Figure 60, Figure 61, and Figure 62 show the same plots as described before, but for heavy aircraft. Here, there is a more noticeable difference between the two models in the DFW location as well. This suggests that as aircraft weight increases, the difference between the two models becomes more apparent.

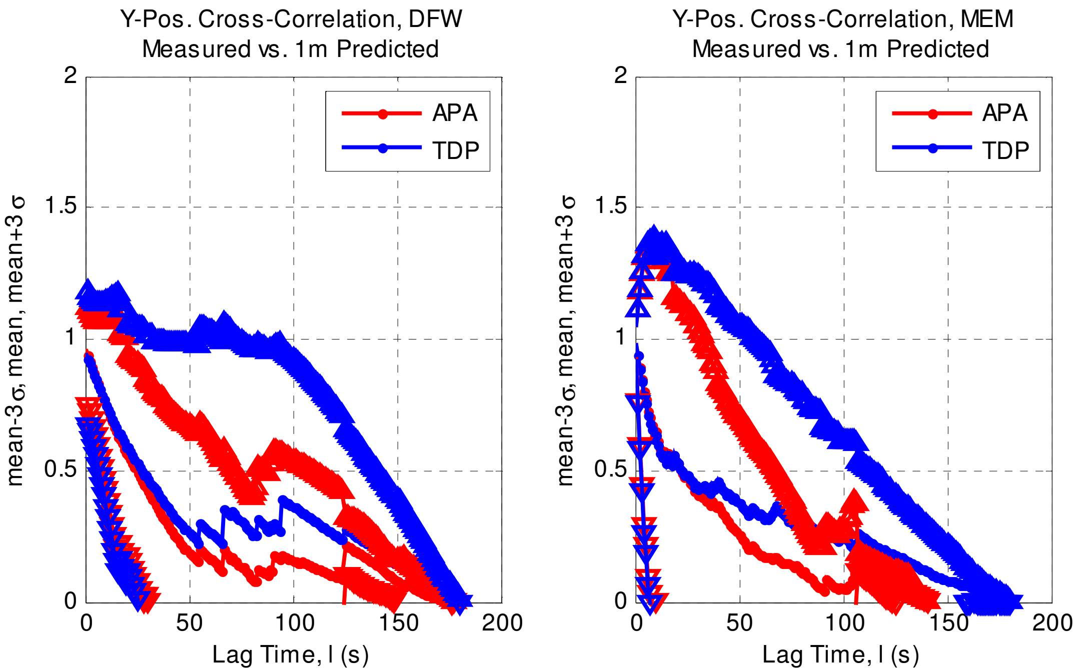


FIGURE 61. MEAN Y-POSITION CROSS-CORRELATION OF MEASURED AND 1M RESOLUTION PREDICTED DATA FOR **HEAVY** AIRCRAFT. UPPER AND LOWER BOUNDS (TRIANGLES) SHOW +/- 3 STANDARD DEVIATIONS FROM MEAN (POINTS).

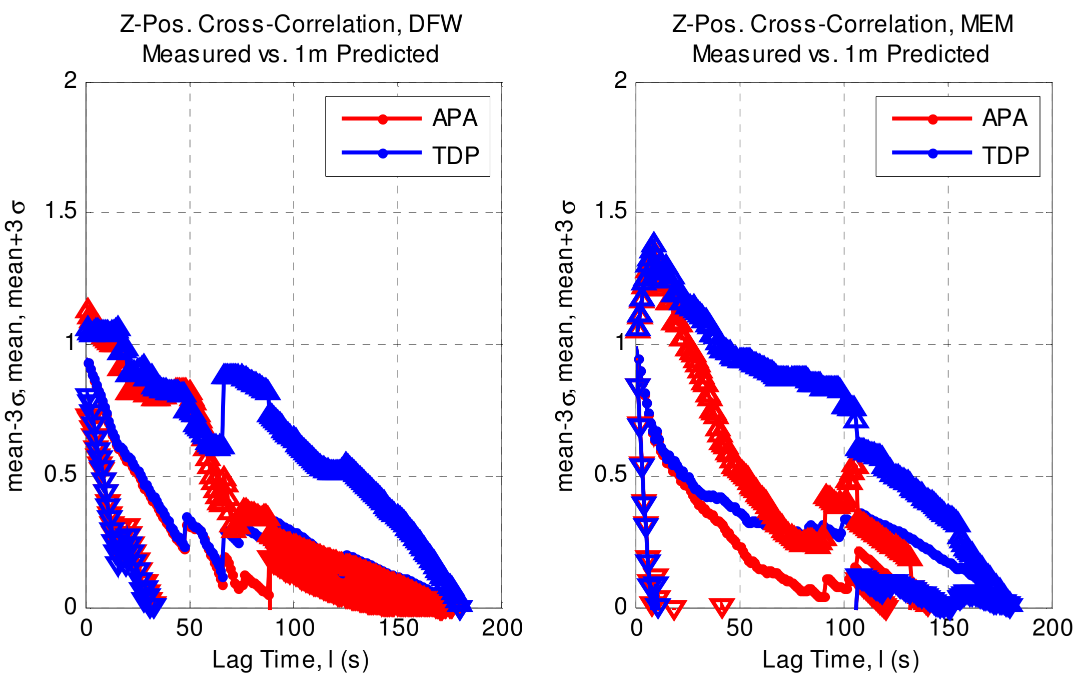


FIGURE 62. MEAN Z-POSITION CROSS-CORRELATION OF MEASURED AND 1M RESOLUTION PREDICTED DATA FOR **HEAVY** AIRCRAFT. UPPER AND LOWER BOUNDS (TRIANGLES) SHOW +/- 3 STANDARD DEVIATIONS FROM MEAN (POINTS).

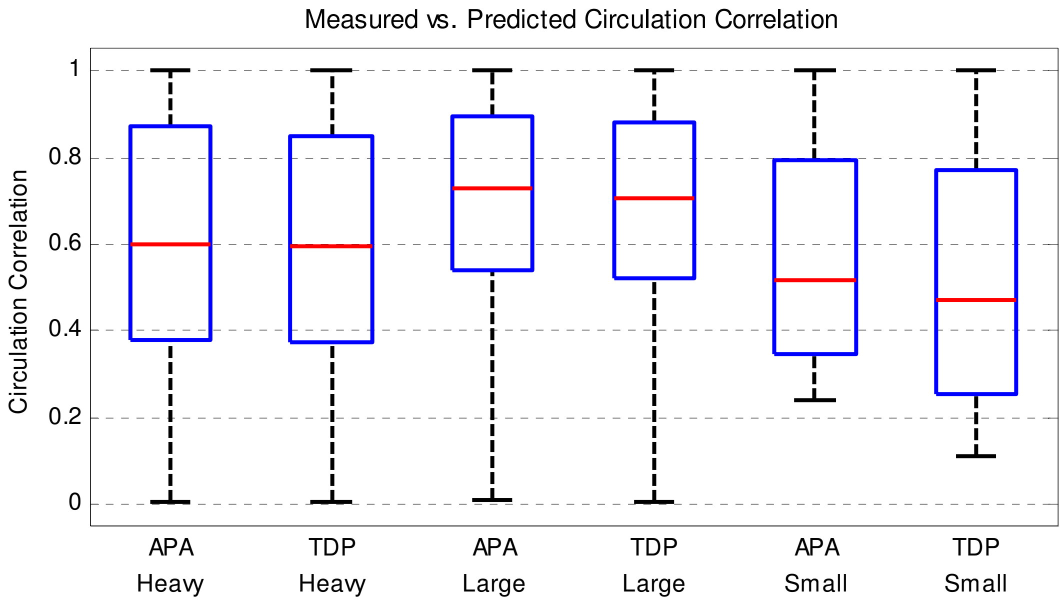


FIGURE 63. MEASURED VS. PREDICTED CIRCULATION CORRELATION (LAG TIME = 0) DISTRIBUTIONS INCLUDING OUTLIERS.

The cross-correlation function is more applicable when the lag time is set to zero. When this is done, the function becomes a correlation function. Figure 63, Figure 64, and Figure 65 show the results of this for circulation, Y-position, and Z-position for all the data available. The box plots show the distribution of the correlation coefficients for each pair of measured and predicted data, with outliers included.

The mean circulation correlation seems to be maximum for the large aircraft case, but this is most likely due to the large number of large aircraft cases available. There were slightly fewer heavy aircraft cases and practically no small aircraft cases. The heavy aircraft cases also had more outliers than the large aircraft cases, further pulling down the correlation.

The Y-position and Z-position results show much higher mean correlation between measured and predicted data across all aircraft times, wake models, and locations. These numerical means can be used as a basis for scoring models based on correlation. However, it is apparent that there is very little difference between the models in terms of correlation.

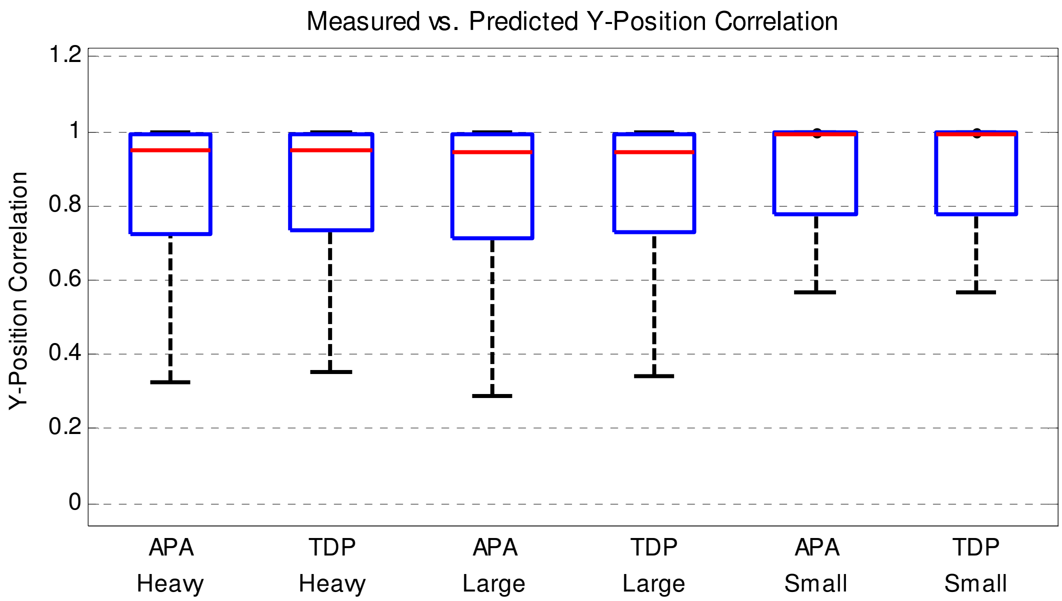


FIGURE 64. MEASURED VS. PREDICTED Y-POSITION CORRELATION (LAG TIME = 0) DISTRIBUTIONS INCLUDING OUTLIERS.

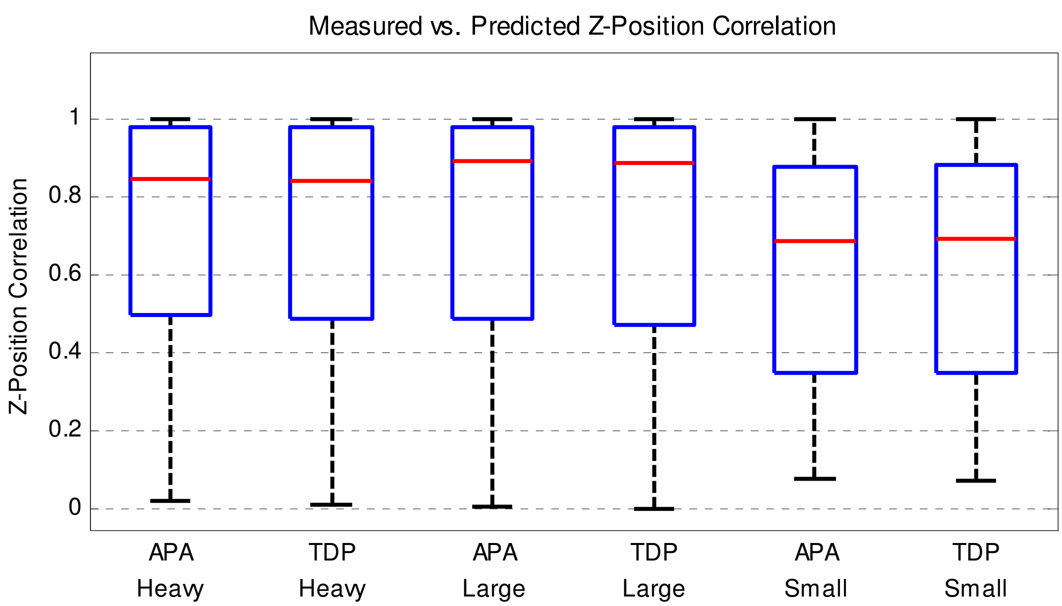


FIGURE 65. MEASURED VS. PREDICTED Z-POSITION CORRELATION (LAG TIME = 0) DISTRIBUTIONS INCLUDING OUTLIERS.

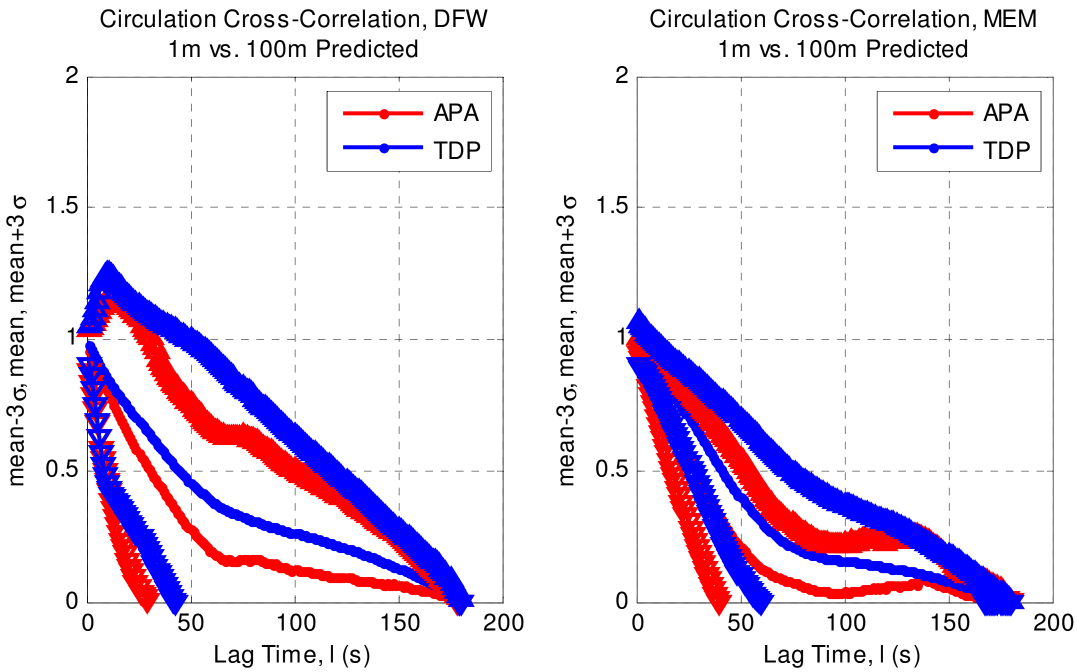


FIGURE 66. MEAN CIRCULATION CROSS-CORRELATION OF 1M RESOLUTION AND 100M RESOLUTION PREDICTED DATA FOR **LARGE** AIRCRAFT. UPPER AND LOWER BOUNDS (TRIANGLES) SHOW +/- 3 STANDARD DEVIATIONS FROM MEAN (POINTS).

Figure 66, Figure 67, and Figure 68 show the mean cross-correlation of 1m height resolution and 100m height resolution predicted data for large aircraft for circulation, Y-position, and Z-position, respectively. This is the cross-correlation of the lowest resolution predicted data with the highest resolution predicted data. The TDP model maintains better correlation between these two datasets for larger lag times. This indicates that the TDP results vary less with time than the APA results.

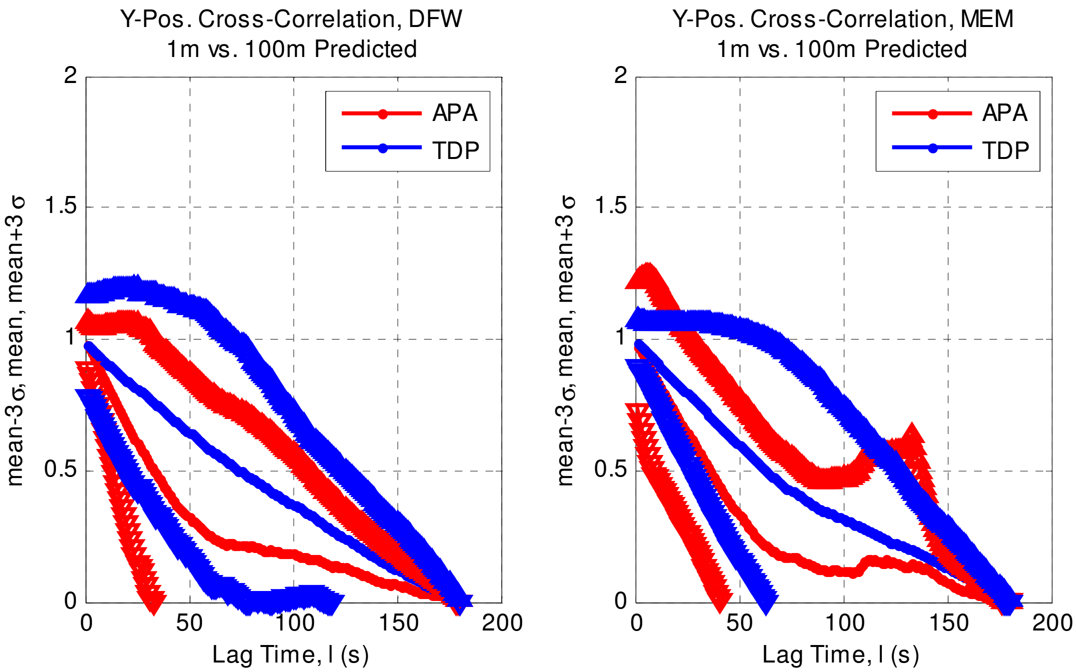


FIGURE 67. MEAN Y-POSITION CROSS-CORRELATION OF 1M RESOLUTION AND 100M RESOLUTION PREDICTED DATA FOR **LARGE** AIRCRAFT. UPPER AND LOWER BOUNDS (TRIANGLES) SHOW +/- 3 STANDARD DEVIATIONS FROM MEAN (POINTS).

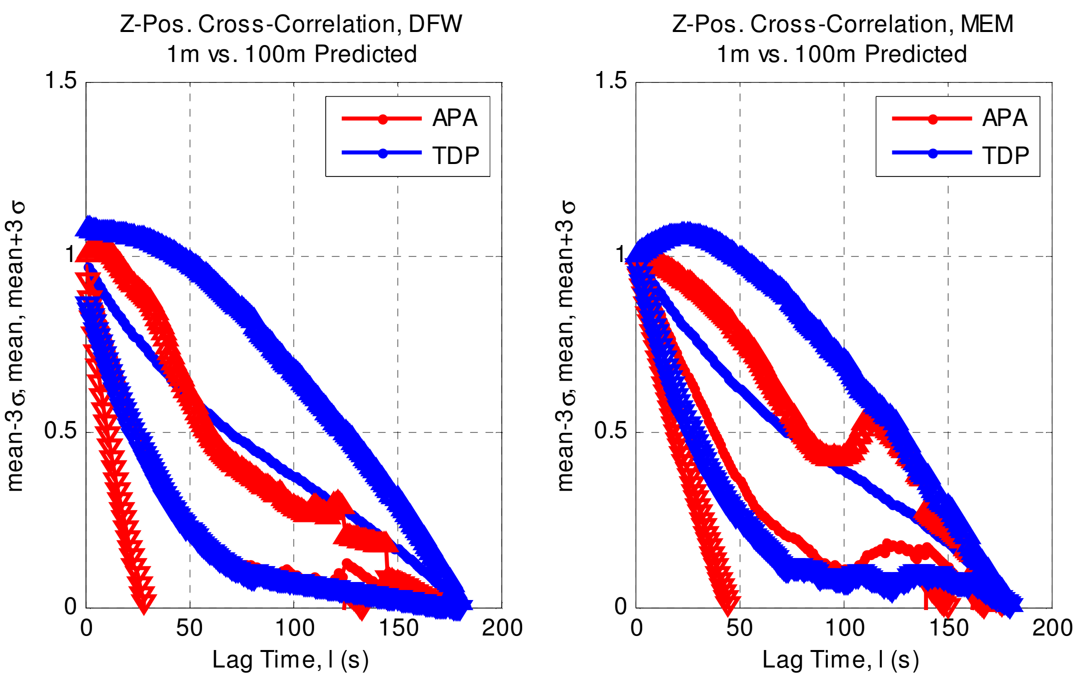


FIGURE 68. MEAN Z-POSITION CROSS-CORRELATION OF 1M RESOLUTION AND 100M RESOLUTION PREDICTED DATA FOR **LARGE** AIRCRAFT. UPPER AND LOWER BOUNDS (TRIANGLES) SHOW +/- 3 STANDARD DEVIATIONS FROM MEAN (POINTS).

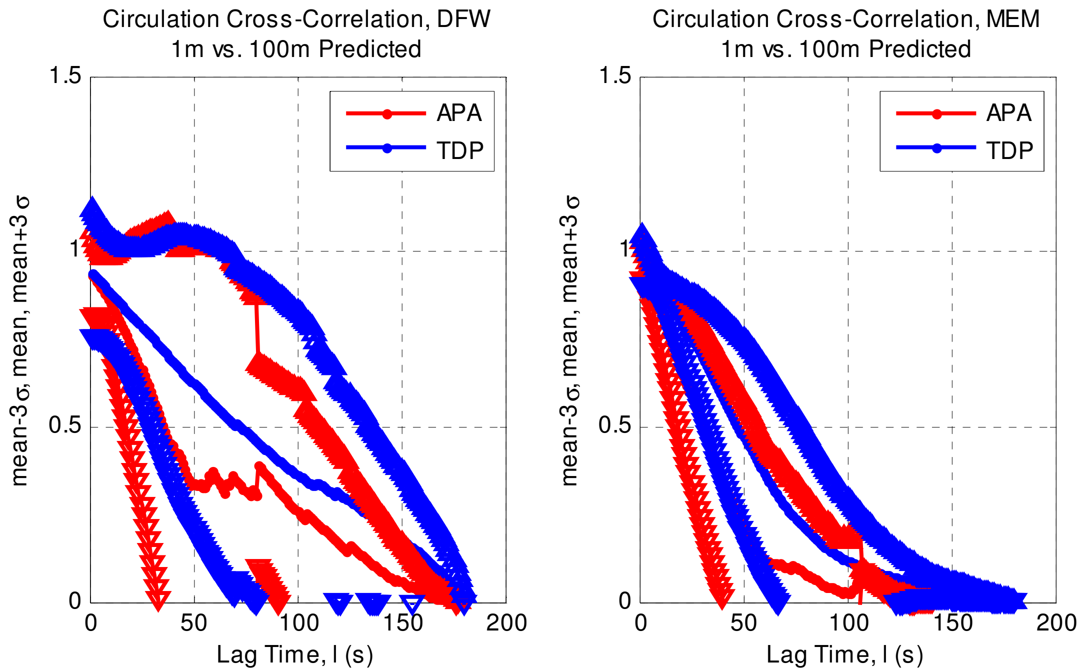


FIGURE 69. MEAN CIRCULATION CROSS-CORRELATION OF 1M RESOLUTION AND 100M RESOLUTION PREDICTED DATA FOR **HEAVY** AIRCRAFT. UPPER AND LOWER BOUNDS (TRIANGLES) SHOW +/- 3 STANDARD DEVIATIONS FROM MEAN (POINTS).

The same can be seen for the heavy aircraft case, in Figure 69, Figure 70, and Figure 71, respectively. Here, the behavior is more pronounced, with the difference in mean cross-correlation between the APA and TDP models more significant. This indicates that the differences between the models only appear when the vortices are generated by heavier aircraft. This may be due to the wake vortex being stronger and more pronounced behind heavier aircraft. Reducing the meteorological data height resolution would be less of a detriment to the prediction accuracy for heavier aircraft because the strong vortices are less susceptible to changes in meteorological conditions.

In Task 1, it was observed that the TDP model showed a smaller variation in average precision as resolution was decreased than the APA model. The present analysis shows within the time domain, this is also true. That is, not only does the average precision change very little across several resolutions, but the precision from time step to time step also does not change significantly between resolutions.

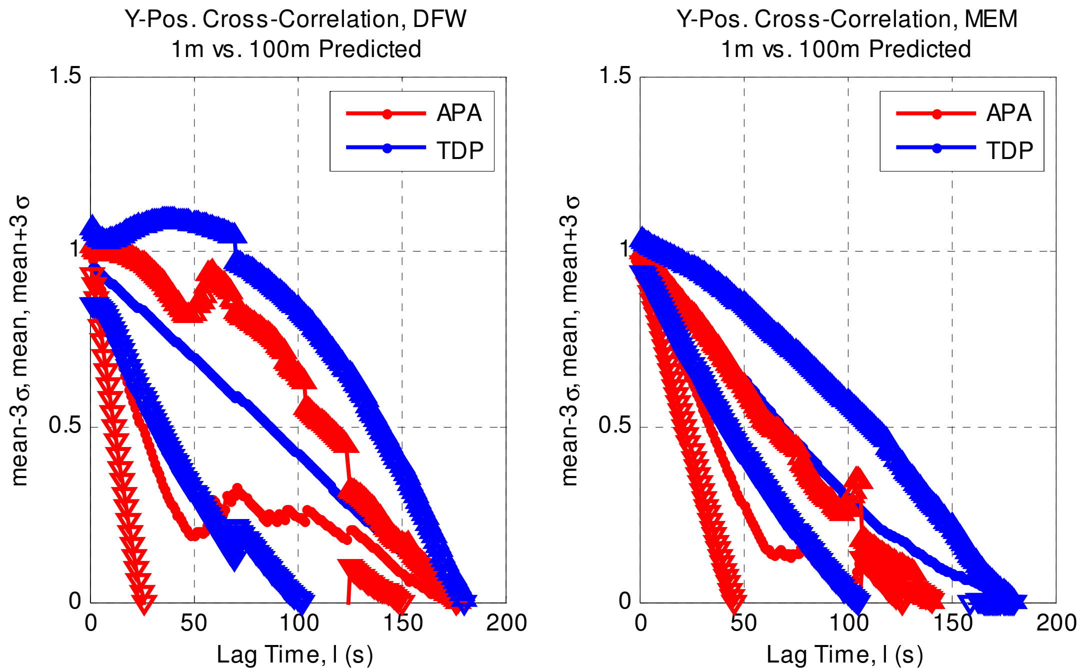


FIGURE 70. MEAN Y-POSITION CROSS-CORRELATION OF 1M RESOLUTION AND 100M RESOLUTION PREDICTED DATA FOR **HEAVY** AIRCRAFT. UPPER AND LOWER BOUNDS (TRIANGLES) SHOW +/- 3 STANDARD DEVIATIONS FROM MEAN (POINTS).

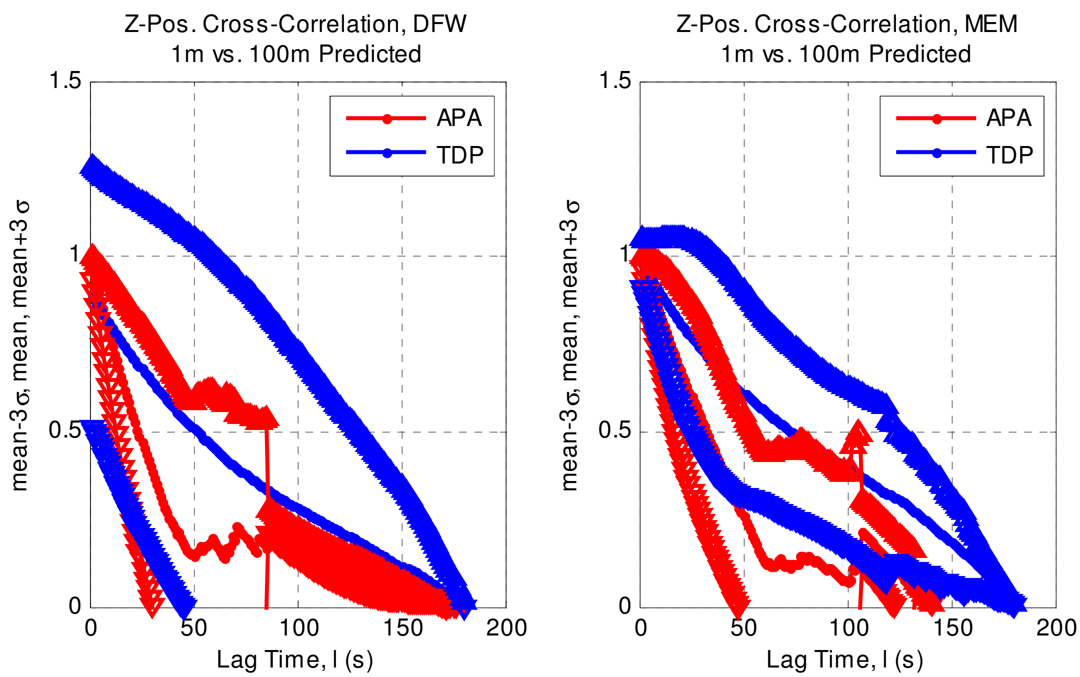


FIGURE 71. MEAN Z-POSITION CROSS-CORRELATION OF 1M RESOLUTION AND 100M RESOLUTION PREDICTED DATA FOR **HEAVY** AIRCRAFT. UPPER AND LOWER BOUNDS (TRIANGLES) SHOW +/- 3 STANDARD DEVIATIONS FROM MEAN (POINTS).

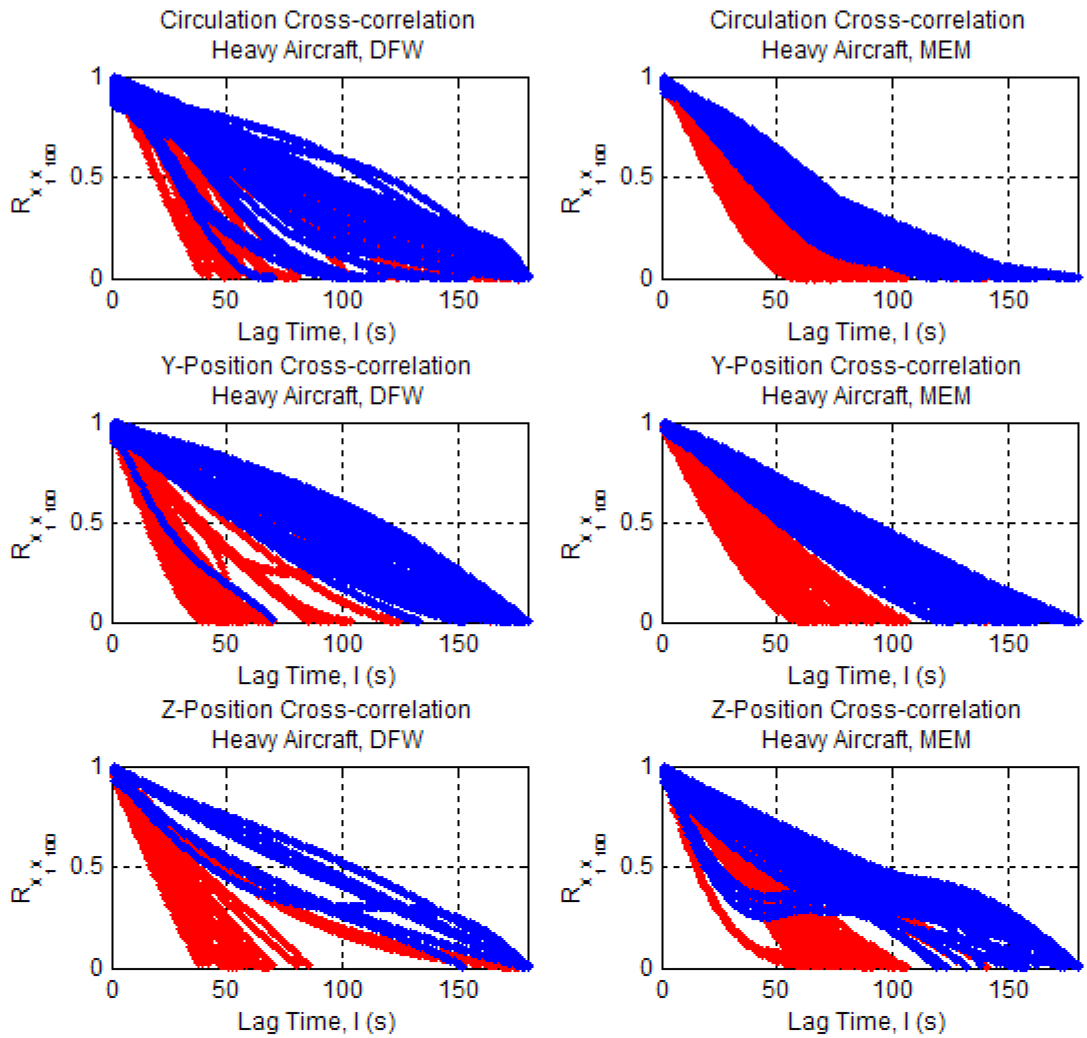


FIGURE 72. CROSS-CORRELATION FOR 1M RESOLUTION PREDICTED DATA AND 100M RESOLUTION PREDICTED DATA; HEAVY AIRCRAFT; KEY: APA, TDP

To illustrate how dramatic the difference between APA and TDP is when considering the cross-correlation of the highest and lowest resolution data, the full data set for heavy aircraft is plotted in Figure 72. From this, it is readily evident that the TDP model maintains higher correlation than the APA model at higher lag times.

SUMMARY

Three methods for scoring wake vortex models have been presented: a residual method, a corridor exit time method, and a cross-correlation method. Each method can provide insight into specific areas of strength and weakness of wake models.

The residual method provides the most thorough scoring of model accuracy, as it compares model output point by point with experimental data. Both the APA and TDP methods perform similarly under this method, agreeing with results observed in the sensitivity study performed in Task 1. The corridor exit method also shows very little difference between the methods. Both models produce a very conservative estimate of the vortex exit time from the defined corridor. At a smaller corridor size however, the TDP method significantly under-predicts corridor exit time for heavy aircraft.

The cross-correlation method provides a measure of the uncertainty between the measured and predicted data with time. It can be used to align measured and predicted data by finding the lag time with the highest correlation. Under this methodology, the TDP method retained correlation better than the APA method did with time, suggesting that the TDP method is more resistant to changes in circulation and vortex position than the APA method as time goes on.

REFERENCES

1. Switzer, G. P. "APA Suite Version 4.19 Users Guide." Analytical Services and Materials, Inc., 2009.
2. Campbell, S. D., Dasey, T. J., Freehart, R. E., Heinrichs, R. M., Mathews, M. P., Perras, G. H., and Rowe, G. S. "Wake Vortex Field Measurement Program at Memphis, Tennessee: Data Guide." Langley Research Center, Hampton, VA, 1997.
3. "National Environmental Satellite, Data, and Information Service (NESDIS)." Vol. 2010, 2010.
4. Proctor, F. H., and Hamilton, D. W. "Evaluation of Fast-Time Wake Vortex Prediction Models," *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*. Orlando, FL, 2009.
5. "Memphis Data Set User Information." 1997.
6. Sarpkaya, T., Robins, R. E., and Delisi, D. P. "Wake-Vortex Eddy-Dissipation Model Predictions Compared with Observations," *Journal of Aircraft* Vol. 38, No. 4, 2001, pp. 687-692.

doi: 10.2514/2.2820

7. "NASA Langley Research Center AVOSS 1999 DFW Deployment Meterological and Wake Vortex Data November 10 - December 3." 1a ed., Hampton, VA, 2000.
8. Hallock, J. N. "Aircraft Wake Vortices: An Annotated Bibliography (1923-1990)." John A. Volpe National Transportation Systems Center, Cambridge, MA, 1991.
9. Kraft, C. C., Jr. "Flight Measurements of the Velocity Distribution and Persistence of the Trailing Vortices of an Airplane." Langley Research Center, Hampton, VA, 1955.
10. McGowan, W. A. "Calculated Normal Load Factors on Light Airplanes Traversing the Trailing Vortices of Heavy Transport Airplanes." Langley Research Center, Hampton, VA, 1961.
11. Reeder, J. P., and Wetmore, J. W. "Aircraft Vortex Wakes in Relation to Terminal Operations." Langley Research Center, Hampton, VA, 1963.

12. Robinson, G. H., and Larson, R. R. "A Flight Evaluation of Methods for Predicting Vortex Wake Effects on Trailing Aircraft." Dryden Flight Research Center, Edwards, CA, 1972.
13. Crow, S. C., and E. R. Bate, J. "Lifespan of Trailing Vortices in a Turbulent Atmosphere," *Journal of Aircraft* Vol. 13, No. 7, 1976, pp. 476-482.

doi: 10.2514/3.44537
14. Greene, G. C. "An Approximate Model of Vortex Decay in the Atmosphere," *Journal of Aircraft* Vol. 23, No. 7, 1986, pp. 566-573.

doi: 10.2514/3.45345
15. Kantha, L. H. "Empirical Model of Transport and Decay of Wake Vortices," *Journal of Aircraft* Vol. 35, No. 4, 1998, pp. 649-653.

doi: 10.2514/2.2350
16. Kantha, L. H. "Empirical Model of Transport and Decay of Wake Vortices Between Parallel Runways," *Journal of Aircraft* Vol. 33, No. 4, 1996, pp. 752-760.

doi: 10.2514/3.47011
17. Corjon, A., and Poinsot, T. "Vortex Model to Define Safe Aircraft Separation Distances," *Journal of Aircraft* Vol. 33, No. 3, 1996, pp. 547-553.

doi: 10.2514/3.46979
18. Liu, H. "Tow-Tank Simulation of Vortex Wake Dynamics," *FAA Proceedings of the Aircraft Wake Vortices Conference*. Washington, DC, 1991, pp. 32-1 - 32-26.
19. Sarpkaya, T. "New Model for Vortex Decay in the Atmosphere," *Journal of Aircraft* Vol. 37, No. 1, 2000, pp. 53-61.

doi: 10.2514/2.2561
20. Robins, R. E., Delisi, D. P., and Greene, G. C. "Development and Validation of a Wake Vortex Predictor Algorithm," *36th Aerospace Sciences Meeting and Exhibit*. Reno, NV, 1998.
21. Zheng, Z. C., and Lim, S. H. "Validation and Operation of a Wake Vortex/Shear Interaction Model," *Journal of Aircraft* Vol. 37, No. 6, 2000, pp. 1073-1078.

doi: 10.2514/2.2713
22. Robins, R. E., and Delisi, D. P. "NWRA AVOSS Wake Vortex Prediction Algorithm." Langley Research Center, Hampton, VA, 2002.
23. Delisi, D. P., Robins, R. E., Switzer, G. F., Lai, D. Y., and Wang, F. Y. "Comparison of Numerical Model Simulations and SFO Wake Vortex Windline Measurements," *21st AIAA Applied Aerodynamics Conference*. Orlando, FL, 2003.
24. Proctor, F. H. "The Terminal Area Simulation System. Volume 1: Theoretical Formulation." NASA, Washington, D.C., 1987.
25. Proctor, F. H. "The Terminal Area Simulation System. Volume 2: Verification Cases." NASA, Washington, D.C., 1987.
26. Switzer, G. F. "Validation Tests of TASS for Application to 3-D Vortex Simulations." Langley Research Center, Hampton, VA, 1996.
27. Proctor, F. H., Hamilton, D. W., and Switzer, G. "TASS Driven Algorithms for Wake Prediction," *44th AIAA Aerospace Sciences Meeting and Exhibit*. Reno, NV, 2006.
28. Holzäpfel, F. "Probabilistic Two-Phase Wake Vortex Decay and Transport Model," *Journal of Aircraft* Vol. 40, No. 2, 2003, pp. 323-331.

doi: 10.2514/2.3096

29. Holzäpfel, F. "Probabilistic Two-Phase Aircraft Wake-Vortex Model: Further Development and Assessment," *Journal of Aircraft* Vol. 43, No. 3, 2006, pp. 700-708.

doi: 10.2514/1.16798

30. Robins, R. E., and Delisi, D. P. "Wake Vortex Algorithm Scoring Results." Langley Research Center, 2002.

31. Hinton, D. A. "An Aircraft Vortex Spacing System (AVOSS) For Dynamical Wake Vortex Spacing Criteria," *AGARD 78th Fluid Dynamics Panel Meeting and Symposium*. Trondheim, Norway, 1996.

32. Keane, R. D., and Adrian, R. J. "Theory of Cross-Correlation Analysis of PIV Images," *Applied Scientific Research* Vol. 49, No. 3, 1992, pp. 191-215.

APPENDICES

APPENDIX A: ACCOMPANYING DATA FILES

Folder	Subfolders/Files	Contains
Aircraft Data		Information on test cases by aircraft and run
	BY_AIRCR.xls	Supplied aircraft type, altitude, landing speed, weight, wingspan, and theoretical calculation of initial vortex strength for each measured vortex case number from MEM
	Aircraft Data.xls	Data by aircraft type
AVOSS DFW99		Weather conditions for test runs completed at DFW
	Daily_reports.pdf	Supplied weather summaries by day
	Wind_clouds99.pdf	Supplied weather categories and stability classes
Code		
	Matlab Parsing and Execution Script	Code to run parsing and execution script written in Matlab APAparse_7.m is the script to run. This calls all other scripts and databases in the folder.
	• APAparse_7.m – code to parse and execute APA Suite • importATMPRO.m – imports DFW atmpro files • importEDATA.m – imports DFW EDATA files • importEDR.m – imports MEM EDR files • importTDATA.m – imports TDATA files • importUT.m – imports MEM U and T data from files • importVDB.m – imports vortex database • VortexDB2.txt – contains case numbers for vortex files • VortexDB.xlsx – database containing dates and times of vortex measurements	
	Post-Processing Scripts	Files to run post-processing scripts

	• dataanalysis2.m – reads MeasureRMSCompData20100519v2.xlsx, generates .mat containing saved workspace information, makes histograms and boxplots, completes ANOVA analysis for measured vs predicted data • getrms.m – computes the normalized mean RMS of the vortex strength, Y-position, and Z-position • intramodelRMS.mat – Matlab workspace generated by makeAnovaMat.m • makeANOVAmat.m – reads Results_intraRMS_20100625.xls, generates .mat containing intramodel RMS data in a readable format by MATLAB's ANOVA functions • Measured vs. Predicted.mat – Matlab workspace generated by dataanalysis2.m • plotdata.m – uses intramodelRMS.mat, makes box plots, performs ANOVA analysis, creates histograms of data • rms2xls.m – reads RMS .mat files and writes .xls files • rmscalc.m - stores all the measured and predicted vortex time series data into .mat files and calls getrms.m • writeall.m – combines RMS.mat files, measured vortex .mat files, and predicted vortex .mat files into single spreadsheet called "Results_All_20100507.xls"	
Databases and Excel Sheets		Databases and Excel files created during research work
	• MeasureRMSCompData20100519v2.xlsx – measured vs. predicted RMS data • Meteorological Database.xls – database of meteorological information • Meteorological Data ToC.xls – catalogue of supplied meteorological data • Results_All_20100507.xls – database of results as of 05/07/2010 • Results_All_Avg_20100625.xls – database of average results as of 06/25/2010 • Results_intramodelRMS_20100625.xls – intramodel RMS database of results • Results – Measured Vortices.xls – measured vortices recorded at DFW and MEM • Results – Predicted Vortices_DFW_only.xls – predicted vortex results from DFW • Results – Predicted Vortices_MEM_only.xls – predicted vortex results from MEM • Vortex Database.xls – vortex database containing vortex data	
Input Files for APA Suite		Data files (ACDATA, QDATA, TDATA, and UDATA) at original and interpolated resolutions to use for APA Suite execution
Output Files for APA Suite		Data files (APA.out and TRAJEC.dat) from APA Suite execution
References for Task 1 Report		All documents referenced in this report

APPENDIX B: NUMERICAL DATA TABLES

Appendix B contains tables summarizes the cases that were simulated and numerical summaries of all the box plots contained in the report.

TABLE 23. SUMMARY OF SIMULATED CASES.

Categories		Number of Cases
Total Cases		2895
Measured Data Properties	**D = Displace Only**	1861
	T = Truncate and Displace	243
	E = Single Data Point (Empty)	791
Analysis Type	**APA**	1456
	TDP	1439
Location	**DFW**	1272
	MEM	1623
Aircraft Weight Class	**Small**	10
	Large	2521
	Heavy	364
Winds	**Calm**	1242
	Windy	1303
	Unknown	350
Cloud Cover	**Clear**	2231
	Cloudy	314
	Unknown	350

TABLE 24. SUMMARY OF OUTLIERS.

			Number of Cases
Measured vs. Predicted Analysis	**Total Cases**		2104
	Outliers	RMS_{circ}	101
		RMS_{Y}	6
		RMS_{Z}	26
Intramodel Analysis	**Total Cases (including all resolutions)**		23160
	Outliers	RMS_{circ}^{norm}	1519
		RMS_{Y}^{norm}	5
		RMS_{Z}^{norm}	2360

TABLE 25. DATA SUMMARY FOR FIGURE 5.

1m Resolution	**Outliers Included**		**Outliers Excluded**	
	MSE_{circ}^{norm} (m^2/s)		**MSE_{circ}^{norm} (m^2/s)**	
Maximum Outlier	15.6243		N/A	
Upper Whisker	2.8821		2.8821	
Upper Quartile	1.9254		1.9144	
Median	1.6262		1.6439	
Lower Quartile	1.2321		1.3316	
Lower Whisker	0.1942		0.1596	
Minimum Outlier	0.0325		N/A	
	MSE_Y^{norm} (m)	**MSE_Z^{norm} (m)**	**MSE_Y^{norm} (m)**	**MSE_Z^{norm} (m)**
Maximum Outlier	496.8435	6.0596	N/A	N/A
Upper Whisker	3.7116	0.8848	29.0584	0.7234
Upper Quartile	1.6226	0.4780	1.5965	0.4729
Median	0.5356	0.2865	0.5245	0.2742
Lower Quartile	0.2222	0.1399	0.2203	0.1361
Lower Whisker	0.0148	0.0282	0.0148	0.0282
Minimum Outlier	None	None	N/A	N/A

TABLE 26. DATA SUMMARY FOR FIGURE 7.

100m Resolution	**Outliers Included**		**Outliers Excluded**	
	MSE_{circ}^{norm} (m^2/s)		**MSE_{circ}^{norm} (m^2/s)**	
Maximum Outlier	15.6270		N/A	
Upper Whisker	2.8492		2.8707	
Upper Quartile	1.9231		1.9179	
Median	1.6171		1.6356	
Lower Quartile	1.2938		1.3519	
Lower Whisker	0.3513		0.1719	
Minimum Outlier	0.0447		N/A	
	MSE_Y^{norm} (m)	**MSE_Z^{norm} (m)**	**MSE_Y^{norm} (m)**	**MSE_Z^{norm} (m)**
Maximum Outlier	525.7459	6.0598	N/A	N/A
Upper Whisker	3.9457	1.0022	26.3737	0.7366
Upper Quartile	1.7168	0.4851	1.7013	0.4810
Median	0.5188	0.2865	0.5155	0.2746
Lower Quartile	0.2186	0.1367	0.2184	0.1354
Lower Whisker	0.0146	0.0275	0.0146	0.0275
Minimum Outlier	None	None	N/A	N/A

TABLE 27. DATA SUMMARY FOR FIGURE 9.

1m vs. 100m Resolution	1m		100m	
	MSE_{circ}^{norm} (m^2/s)		MSE_{circ}^{norm} (m^2/s)	
Upper Whisker	2.8821		2.8707	
Upper Quartile	1.9144		1.9179	
Median	1.6439		1.6356	
Lower Quartile	1.3316		1.3519	
Lower Whisker	0.1596		0.1719	
	MSE_Y^{norm} (m)	MSE_Z^{norm} (m)	MSE_Y^{norm} (m)	MSE_Z^{norm} (m)
Upper Whisker	29.0584	0.7234	26.3737	0.7366
Upper Quartile	1.5965	0.4729	1.7013	0.4810
Median	0.5245	0.2742	0.5155	0.2746
Lower Quartile	0.2203	0.1361	0.2184	0.1354
Lower Whisker	0.0148	0.0282	0.0146	0.0275

TABLE 28.DATA SUMMARY FOR FIGURE 10.

	MSE_{circ}^{norm} (m^2/s)	
	Location	
	DFW	MEM
Upper Whisker	2.8821	2.8493
Upper Quartile	1.7413	2.0133
Median	1.3377	1.7749
Lower Quartile	0.4555	1.5853
Lower Whisker	0.1596	1.1048

TABLE 29. DATA SUMMARY FOR FIGURE 11.

	MSE_Y^{norm} (m)						
	Location		Aircraft Weight Class			Winds	
	DFW	MEM	Heavy	Large	Small	Calm	Windy
Upper Whisker	29.0584	10.2445	29.0584	22.4651	0.8011	29.0584	19.2976
Upper Quartile	3.1761	0.5539	2.6968	1.4984	0.4346	1.1533	1.3144
Median	1.2682	0.2681	0.8123	0.4924	0.0416	0.4099	0.4913
Lower Quartile	0.4850	0.1646	0.2437	0.2176	0.0149	0.2056	0.2238
Lower Whisker	0.0484	0.0148	0.0769	0.0445	0.0148	0.0148	0.0445

TABLE 30. DATA SUMMARY FOR FIGURE 12.

	MSE_Y^{norm} (m)					
	Cloud Cover		Winds and Cloud Cover			
	Clear	Cloudy	Calm & Clear	Calm & Cloudy	Windy & Clear	Windy & Cloudy
Upper Whisker	29.0584	15.3636	29.0584	9.6245	19.2976	15.3636
Upper Quartile	1.2824	1.0690	1.2581	0.3983	1.2914	1.4991
Median	0.4664	0.3880	0.4516	0.3394	0.4888	0.5725
Lower Quartile	0.2034	0.2394	0.1997	0.2245	0.2072	0.2432
Lower Whisker	0.0148	0.0445	0.0148	0.0955	0.0484	0.0445

TABLE 31. DATA SUMMARY FOR FIGURE 13.

	MSE_Z^{norm} (m)			
	Location		Winds	
	DFW	MEM	Calm	Windy
Upper Whisker	0.7211	0.7234	0.7234	0.7147
Upper Quartile	0.5389	0.2815	0.4646	0.3695
Median	0.4474	0.2139	0.2671	0.2316
Lower Quartile	0.2683	0.1027	0.1373	0.1179
Lower Whisker	0.0412	0.0282	0.0291	0.0282

TABLE 32. DATA SUMMARY FOR FIGURE 14.

	MSE_Z^{norm} (m)					
	Cloud Cover		Winds and Cloud Cover			
	Clear	Cloudy	Calm, Clear	Calm, Cloudy	Windy, Clear	Windy, Cloudy
Upper Whisker	0.7234	0.5956	0.7234	0.5956	0.7147	0.5900
Upper Quartile	0.4553	0.2713	0.4658	0.1579	0.3793	0.3041
Median	0.2592	0.1240	0.2954	0.0592	0.2440	0.1710
Lower Quartile	0.1402	0.0585	0.1764	0.0582	0.1212	0.0619
Lower Whisker	0.0291	0.0282	0.0291	0.0573	0.0346	0.0282

TABLE 33. DATA SUMMARY FOR FIGURE 17.

5m Resolution 30 Sec. Avg.	**Outliers Included**		**Outliers Excluded**	
	RMS_{circ}^{norm}		**RMS_{circ}^{norm}**	
Maximum Outlier	1.0604		N/A	
Upper Whisker	1.0000		1.0055	
Upper Quartile	1.0000		1.0000	
Median	1.0000		1.0000	
Lower Quartile	1.0000		1.0000	
Lower Whisker	1.0000		0.9947	
Minimum Outlier	0.9533		N/A	
	RMS_{Y}^{norm}	**RMS_{Z}^{norm}**	**RMS_{Y}^{norm}**	**RMS_{Z}^{norm}**
Maximum Outlier	1.0967	1.0290	N/A	N/A
Upper Whisker	1.0001	1.0002	1.0054	1.0052
Upper Quartile	1.0000	1.0000	1.0000	1.0000
Median	1.0000	1.0000	1.0000	1.0000
Lower Quartile	0.9999	0.9999	1.0000	0.9999
Lower Whisker	0.9998	0.9997	0.9946	0.9949
Minimum Outlier	0.7952	0.9713	N/A	N/A

TABLE 34. DATA SUMMARY FOR FIGURE 18.

5m Resolution 60 Sec. Avg.	**Outliers Included**		**Outliers Excluded**	
	RMS_{circ}^{norm}		**RMS_{circ}^{norm}**	
Maximum Outlier	1.1270		N/A	
Upper Whisker	1.0002		1.0098	
Upper Quartile	1.0001		1.0001	
Median	1.0000		1.0000	
Lower Quartile	1.0000		1.0000	
Lower Whisker	0.9999		0.9907	
Minimum Outlier	0.8903		N/A	
	RMS_{Y}^{norm}	**RMS_{Z}^{norm}**	**RMS_{Y}^{norm}**	**RMS_{Z}^{norm}**
Maximum Outlier	1.1586	1.0822	N/A	N/A
Upper Whisker	1.0007	1.0007	1.0054	1.0052
Upper Quartile	1.0002	1.0000	1.0000	1.0000
Median	1.0000	1.0000	1.0000	1.0000
Lower Quartile	0.9998	0.9996	1.0000	0.9999
Lower Whisker	0.9992	0.9990	0.9946	0.9949
Minimum Outlier	0.7946	0.9617	N/A	N/A

TABLE 35. DATA SUMMARY FOR FIGURE 21.

20m Resolution 30 Sec. Avg.	Outliers Included		Outliers Excluded	
	RMS_{circ}^{norm}		RMS_{circ}^{norm}	
Maximum Outlier	1.1066		N/A	
Upper Whisker	1.0011		1.0065	
Upper Quartile	1.0001		1.0001	
Median	1.0000		1.0000	
Lower Quartile	0.9995		0.9995	
Lower Whisker	0.9986		0.9934	
Minimum Outlier	0.9630		N/A	
	RMS_{Y}^{norm}	RMS_{Z}^{norm}	RMS_{Y}^{norm}	RMS_{Z}^{norm}
Maximum Outlier	1.6228	1.0468	N/A	N/A
Upper Whisker	1.0084	1.0012	1.0080	1.0066
Upper Quartile	1.0022	1.0003	1.0013	1.0002
Median	1.0000	1.0000	1.0000	1.0000
Lower Quartile	0.9980	0.9997	0.9988	0.9997
Lower Whisker	0.9918	0.9988	0.9949	0.9935
Minimum Outlier	0.7591	0.9560	N/A	N/A

TABLE 36. DATA SUMMARY FOR FIGURE 22.

20m Resolution 60 Sec. Avg.	Outliers Included		Outliers Excluded	
	RMS_{circ}^{norm}		RMS_{circ}^{norm}	
Maximum Outlier	1.1536		N/A	
Upper Whisker	1.0039		1.0148	
Upper Quartile	1.0006		1.0006	
Median	0.9998		0.9998	
Lower Quartile	0.9984		0.9986	
Lower Whisker	0.9951		0.9852	
Minimum Outlier	0.8365		N/A	
	RMS_{Y}^{norm}	RMS_{Z}^{norm}	RMS_{Y}^{norm}	RMS_{Z}^{norm}
Maximum Outlier	4.2523	1.1488	N/A	N/A
Upper Whisker	1.0186	1.0039	1.0080	1.0066
Upper Quartile	1.0051	1.0009	1.0013	1.0002
Median	1.0000	1.0000	1.0000	1.0000
Lower Quartile	0.9961	0.9988	0.9988	0.9997
Lower Whisker	0.9825	0.9958	0.9949	0.9935
Minimum Outlier	0.4532	0.8651	N/A	N/A

TABLE 37. DATA SUMMARY FOR FIGURE 25.

100m Resolution 30 Sec. Avg.	Outliers Included		Outliers Excluded	
	RMS_{circ}^{norm}		RMS_{circ}^{norm}	
Maximum Outlier	1.0618		N/A	
Upper Whisker	1.0310		1.0527	
Upper Quartile	0.9999		1.0000	
Median	0.9939		0.9948	
Lower Quartile	0.9788		0.9826	
Lower Whisker	0.9474		0.9024	
Minimum Outlier	0.4638		N/A	
	RMS_{Y}^{norm}	RMS_{Z}^{norm}	RMS_{Y}^{norm}	RMS_{Z}^{norm}
Maximum Outlier	750.2719	1.1442	N/A	N/A
Upper Whisker	1.1038	1.0098	1.5253	1.0658
Upper Quartile	1.0182	1.0036	1.4951	1.0035
Median	0.9937	1.0003	1.4697	1.0003
Lower Quartile	0.9610	0.9994	1.3840	0.9994
Lower Whisker	0.8756	0.9933	1.3802	0.9512
Minimum Outlier	0.0453	0.8902	N/A	N/A

TABLE 38. DATA SUMMARY FOR FIGURE 26.

100m Resolution 60 Sec. Avg.	Outliers Included		Outliers Excluded	
	RMS_{circ}^{norm}		RMS_{circ}^{norm}	
Maximum Outlier	1.1826		N/A	
Upper Whisker	1.0741		1.0762	
Upper Quartile	0.9991		0.9993	
Median	0.9813		0.9828	
Lower Quartile	0.9477		0.9566	
Lower Whisker	0.8708		0.8395	
Minimum Outlier	0.4223		N/A	
	RMS_{Y}^{norm}	RMS_{Z}^{norm}	RMS_{Y}^{norm}	RMS_{Z}^{norm}
Maximum Outlier	514.7000	1.4173	N/A	N/A
Upper Whisker	1.1644	1.0457	1.5253	1.0658
Upper Quartile	1.0266	1.0169	1.4951	1.0035
Median	0.9840	1.0029	1.4697	1.0003
Lower Quartile	0.9346	0.9977	1.3840	0.9994
Lower Whisker	0.7971	0.9690	1.3802	0.9512
Minimum Outlier	0.0172	0.7231	N/A	N/A

TABLE 39. DATA SUMMARY FOR FIGURE 27 AND FIGURE 28.

	RMS_{circ}^{norm} over 30 sec.							
	Height Resolution (m)							
	1 (Unnormalized)	5	15	20	25	30	40	100
Upper Whisker	290.8000	1.0291	1.0372	1.0404	1.0356	1.0404	1.0401	1.0370
Upper Quartile	200.6500	1.0000	1.0002	1.0001	1.0002	1.0002	1.0003	1.0000
Median	166.9800	1.0000	1.0000	1.0000	1.0000	0.9999	0.9998	0.9963
Lower Quartile	107.2500	1.0000	0.9999	0.9995	0.9994	0.9991	0.9982	0.9865
Lower Whisker	36.9780	0.9533	0.9596	0.9630	0.9634	0.9558	0.9554	0.9517
	RMS_{circ}^{norm} over 60 sec.							
Upper Whisker	290.9900	1.0401	1.0403	1.0373	1.0376	1.0400	1.0375	1.0397
Upper Quartile	177.5800	1.0001	1.0007	1.0006	1.0009	1.0009	1.0011	1.0008
Median	143.1300	1.0000	1.0000	0.9998	0.9998	0.9995	0.9990	0.9907
Lower Quartile	99.7630	1.0000	0.9993	0.9984	0.9977	0.9970	0.9945	0.9757
Lower Whisker	37.1400	0.9548	0.9534	0.9535	0.9531	0.9523	0.9526	0.9516

TABLE 40. DATA SUMMARY FOR FIGURE 29 AND FIGURE 30.

	RMS_{Y}^{norm} over 30 sec.							
	Height Resolution (m)							
	1 (Unnormalized)	5	15	20	25	30	40	100
Upper Whisker	745.7600	1.0967	1.7945	1.6228	1.8970	1.9526	1.8967	4.8684
Upper Quartile	211.9500	1.0000	1.0009	1.0022	1.0016	1.0073	1.0039	1.0181
Median	173.1100	1.0000	1.0000	1.0000	0.9994	1.0002	0.9997	0.9937
Lower Quartile	114.7500	0.9999	0.9994	0.9980	0.9954	0.9963	0.9950	0.9610
Lower Whisker	11.1920	0.7952	0.8405	0.7591	0.8369	0.4826	0.7512	0.0453
	RMS_{Y}^{norm} over 60 sec.							
Upper Whisker	607.2200	1.1586	3.0519	4.2523	5.1357	4.3745	5.3466	6.6455
Upper Quartile	185.4000	1.0002	1.0022	1.0051	1.0036	1.0120	1.0073	1.0262
Median	146.6000	1.0000	1.0000	1.0000	0.9988	0.9998	0.9983	0.9840
Lower Quartile	100.4600	0.9998	0.9984	0.9961	0.9928	0.9917	0.9878	0.9346
Lower Whisker	8.6747	0.7946	0.4981	0.4532	0.5776	0.2569	0.4271	0.0172

TABLE 41. DATA SUMMARY FOR FIGURE 31 AND FIGURE 32.

	RMS_Z^{norm} over 30 sec.							
	Height Resolution (m)							
	1 (Unnormalized)	5	15	20	25	30	40	100
Upper Whisker	745.7600	1.0172	1.0195	1.0196	1.0192	1.0188	1.0190	1.0196
Upper Quartile	213.1600	1.0000	1.0003	1.0003	1.0003	1.0004	1.0005	1.0032
Median	172.6400	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0003
Lower Quartile	114.7700	0.9999	0.9996	0.9997	0.9996	0.9995	0.9993	0.9995
Lower Whisker	11.1920	0.9823	0.9826	0.9830	0.9818	0.9819	0.9821	0.9819
	RMS_Z^{norm} over 60 sec.							
Upper Whisker	607.2200	1.0187	1.0196	1.0194	1.0196	1.0195	1.0193	1.0196
Upper Quartile	192.7900	1.0000	1.0007	1.0007	1.0009	1.0012	1.0019	1.0065
Median	148.9600	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0007
Lower Quartile	105.0700	0.9996	0.9991	0.9990	0.9989	0.9987	0.9983	0.9977
Lower Whisker	8.6747	0.9824	0.9819	0.9818	0.9818	0.9817	0.9817	0.9821

TABLE 42. DATA SUMMARY FOR FIGURE 33.

		RMS_{circ}^{norm}				
Height Resolution (m)	**Analysis Method**	**Lower Whisker**	**Lower Quartile**	**Median**	**Upper Quartile**	**Upper Whisker**
100	**TDP**	0.9520	0.9760	0.9899	0.9999	1.0366
	APA	0.9516	0.9740	0.9927	1.0023	1.0397
40	**TDP**	0.9535	0.9963	0.9994	1.0006	1.0352
	APA	0.9526	0.9920	0.9982	1.0020	1.0375
30	**TDP**	0.9523	0.9977	0.9995	1.0001	1.0368
	APA	0.9574	0.9959	0.9994	1.0021	1.0400
25	**TDP**	0.9531	0.9981	0.9998	1.0003	1.0359
	APA	0.9532	0.9971	0.9997	1.0016	1.0376
20	**TDP**	0.9535	0.9988	0.9998	1.0001	1.0354
	APA	0.9579	0.9980	0.9999	1.0016	1.0373
15	**TDP**	0.9534	0.9994	1.0000	1.0003	1.0351
	APA	0.9615	0.9992	1.0000	1.0015	1.0403
5	**TDP**	0.9654	1.0000	1.0000	1.0001	1.0401
	APA	0.9548	1.0000	1.0000	1.0002	1.0352
1 (Unnormalized)	**TDP**	37.1400	105.3213	157.8479	193.5032	289.2115
	APA	38.9004	97.0647	131.7245	160.1135	290.9909

TABLE 43. DATA SUMMARY FOR FIGURE 34.

		RMS_{circ}^{norm}				
Height Resolution (m)	**Location**	**Lower Whisker**	**Lower Quartile**	**Median**	**Upper Quartile**	**Upper Whisker**
100	**MEM**	0.9516	0.9800	0.9937	1.0018	1.0397
	DFW	0.9520	0.9679	0.9823	0.9961	1.0366
40	**MEM**	0.9535	0.9975	0.9998	1.0017	1.0375
	DFW	0.9526	0.9896	0.9964	1.0001	1.0366
30	**MEM**	0.9544	0.9977	0.9998	1.0015	1.0394
	DFW	0.9523	0.9956	0.9989	1.0001	1.0400
25	**MEM**	0.9531	0.9979	0.9999	1.0012	1.0359
	DFW	0.9540	0.9975	0.9995	1.0005	1.0376
20	**MEM**	0.9535	0.9987	0.9999	1.0008	1.0356
	DFW	0.9545	0.9981	0.9996	1.0005	1.0373
15	**MEM**	0.9534	0.9991	0.9999	1.0005	1.0403
	DFW	0.9554	0.9997	1.0000	1.0010	1.0396
5	**MEM**	0.9647	1.0000	1.0000	1.0000	1.0399
	DFW	0.9548	0.9999	1.0001	1.0006	1.0401
1 (Unnormalized)	**MEM**	39.4609	127.0023	155.6878	182.8192	290.5156
	DFW	37.1400	80.8031	113.7018	166.0776	290.9909

TABLE 44. DATA SUMMARY FOR FIGURE 35.

		RMS_{circ}^{norm}				
Height Resolution (m)	**Aircraft Weight Class**	**Lower Whisker**	**Lower Quartile**	**Median**	**Upper Quartile**	**Upper Whisker**
100	**Small**	0.9736	0.9914	0.9966	0.9994	1.0017
	Large	0.9520	0.9746	0.9896	1.0006	1.0397
	Heavy	0.9516	0.9879	0.9953	1.0021	1.0266
40	**Small**	0.9905	0.9961	0.9971	1.0010	1.0300
	Large	0.9526	0.9943	0.9989	1.0011	1.0375
	Heavy	0.9551	0.9960	0.9996	1.0009	1.0350
30	**Small**	0.9923	0.9966	0.9988	1.0033	1.0238
	Large	0.9523	0.9968	0.9994	1.0010	1.0400
	Heavy	0.9667	0.9981	0.9997	1.0007	1.0352
25	**Small**	0.9978	0.9987	1.0000	1.0011	1.0029
	Large	0.9531	0.9976	0.9997	1.0009	1.0359
	Heavy	0.9645	0.9990	1.0000	1.0013	1.0376
20	**Small**	0.9617	0.9969	0.9991	1.0012	1.0059
	Large	0.9535	0.9983	0.9998	1.0006	1.0356
	Heavy	0.9612	0.9991	1.0000	1.0004	1.0373
15	**Small**	0.9732	0.9986	0.9996	1.0002	1.0032
	Large	0.9534	0.9993	1.0000	1.0008	1.0403
	Heavy	0.9655	0.9995	1.0000	1.0005	1.0396
5	**Small**	0.9946	0.9997	1.0000	1.0006	1.0018
	Large	0.9615	1.0000	1.0000	1.0001	1.0399
	Heavy	0.9548	1.0000	1.0000	1.0001	1.0401
1 (Unnormalized)	**Small**	52.5366	69.4828	85.7150	105.1406	133.5611
	Large	38.7752	99.8841	141.2470	173.5919	290.9909
	Heavy	37.1400	100.1962	199.1390	262.5639	290.5156

TABLE 45. DATA SUMMARY FOR FIGURE 36

		RMS_{circ}^{norm}				
Height Resolution (m)	**Winds**	**Lower Whisker**	**Lower Quartile**	**Median**	**Upper Quartile**	**Upper Whisker**
100	**Windy**	0.9516	0.9756	0.9895	1.0000	1.0397
	Calm	0.9520	0.9785	0.9946	1.0018	1.0366
40	**Windy**	0.9528	0.9959	0.9994	1.0018	1.0375
	Calm	0.9526	0.9951	0.9991	1.0009	1.0366
30	**Windy**	0.9523	0.9966	0.9993	1.0010	1.0394
	Calm	0.9580	0.9977	0.9998	1.0010	1.0393
25	**Windy**	0.9531	0.9976	0.9998	1.0011	1.0376
	Calm	0.9532	0.9984	0.9999	1.0007	1.0343
20	**Windy**	0.9535	0.9983	0.9997	1.0006	1.0356
	Calm	0.9545	0.9988	0.9999	1.0006	1.0373
15	**Windy**	0.9534	0.9990	0.9999	1.0005	1.0403
	Calm	0.9554	0.9997	1.0000	1.0008	1.0396
5	**Windy**	0.9643	1.0000	1.0000	1.0000	1.0399
	Calm	0.9548	1.0000	1.0000	1.0001	1.0401
1 (Unnormalized)	**Windy**	38.9004	124.8692	151.7237	178.4894	289.2115
	Calm	37.1400	65.8763	118.6629	174.7968	290.9909

TABLE 46. DATA SUMMARY FOR FIGURE 37

		RMS_{circ}^{norm}				
Height Resolution (m)	**Cloud Cover**	**Lower Whisker**	**Lower Quartile**	**Median**	**Upper Quartile**	**Upper Whisker**
100	**Cloudy**	0.9516	0.9730	0.9900	1.0016	1.0314
	Clear	0.9520	0.9769	0.9914	1.0008	1.0397
40	**Cloudy**	0.9684	0.9936	0.9985	1.0014	1.0348
	Clear	0.9526	0.9956	0.9993	1.0013	1.0375
30	**Cloudy**	0.9591	0.9976	0.9996	1.0012	1.0394
	Clear	0.9523	0.9971	0.9996	1.0010	1.0368
25	**Cloudy**	0.9687	0.9970	0.9995	1.0009	1.0296
	Clear	0.9531	0.9979	0.9999	1.0010	1.0376
20	**Cloudy**	0.9612	0.9983	0.9997	1.0006	1.0354
	Clear	0.9535	0.9986	0.9999	1.0006	1.0373
15	**Cloudy**	0.9615	0.9990	1.0000	1.0006	1.0261
	Clear	0.9534	0.9993	1.0000	1.0006	1.0403
5	**Cloudy**	0.9615	1.0000	1.0000	1.0000	1.0315
	Clear	0.9548	1.0000	1.0000	1.0000	1.0401
1 (Unnormalized)	**Cloudy**	41.9255	61.8799	130.0954	163.0746	272.3892
	Clear	37.1400	101.4555	144.9680	180.0813	290.9909

TABLE 47. DATA SUMMARY FOR FIGURE 38.

		RMS_Y^{norm}				
Height Resolution (m)	**Analysis Method**	**Lower Whisker**	**Lower Quartile**	**Median**	**Upper Quartile**	**Upper Whisker**
100	**TDP**	0.0172	0.9259	0.9828	1.0292	2.8409
	APA	0.2014	0.9409	0.9845	1.0232	6.6455
40	**TDP**	0.4271	0.9865	0.9988	1.0132	5.3466
	APA	0.6884	0.9881	0.9980	1.0037	1.4457
30	**TDP**	0.4640	0.9901	1.0001	1.0176	4.3745
	APA	0.2569	0.9925	0.9997	1.0083	1.4511
25	**TDP**	0.5776	0.9932	1.0000	1.0071	5.1357
	APA	0.8302	0.9926	0.9978	1.0009	1.1233
20	**TDP**	0.4532	0.9945	1.0006	1.0111	4.2523
	APA	0.8372	0.9970	0.9997	1.0025	1.1468
15	**TDP**	0.7385	0.9961	1.0010	1.0092	3.0519
	APA	0.4981	0.9992	1.0000	1.0002	1.0404
5	**TDP**	0.7946	0.9986	1.0000	1.0013	1.1586
	APA	0.8818	1.0000	1.0000	1.0000	1.0253
1 (Unnormalized)	**TDP**	25.4849	113.1258	165.6422	205.6054	607.2246
	APA	8.6747	93.6614	131.7893	161.4992	535.4947

TABLE 48. DATA SUMMARY FOR FIGURE 39.

		RMS_Y^{norm}				
Height Resolution (m)	**Location**	**Lower Whisker**	**Lower Quartile**	**Median**	**Upper Quartile**	**Upper Whisker**
100	**MEM**	0.0172	0.9494	0.9830	1.0053	6.6455
	DFW	0.0790	0.8691	0.9879	1.0866	3.1950
40	**MEM**	0.7771	0.9882	0.9969	1.0016	1.4833
	DFW	0.4271	0.9860	1.0019	1.0205	5.3466
30	**MEM**	0.2569	0.9926	0.9984	1.0017	1.4552
	DFW	0.4640	0.9877	1.0123	1.0392	4.3745
25	**MEM**	0.8810	0.9946	0.9989	1.0016	1.4074
	DFW	0.5776	0.9893	0.9986	1.0093	5.1357
20	**MEM**	0.8815	0.9963	0.9994	1.0011	1.3945
	DFW	0.4532	0.9951	1.0038	1.0164	4.2523
15	**MEM**	0.4981	0.9975	0.9996	1.0011	1.3201
	DFW	0.7385	0.9998	1.0000	1.0080	3.0519
5	**MEM**	0.8841	1.0000	1.0000	1.0002	1.1455
	DFW	0.7946	0.9996	1.0000	1.0001	1.1586
1 (Unnormalized)	**MEM**	24.0748	129.1400	158.0809	198.6652	430.6615
	DFW	8.6747	78.4443	113.9460	170.5723	607.2246

TABLE 49. DATA SUMMARY FOR FIGURE 40.

		RMS_Y^{norm}				
Height Resolution (m)	**Aircraft Weight Class**	**Lower Whisker**	**Lower Quartile**	**Median**	**Upper Quartile**	**Upper Whisker**
100	**Small**	0.5652	0.8902	1.0052	1.0359	1.1838
	Large	0.0172	0.9357	0.9859	1.0313	6.6455
	Heavy	0.1432	0.9236	0.9692	1.0108	2.7864
40	**Small**	0.9128	0.9972	0.9997	1.0087	1.0240
	Large	0.4271	0.9882	0.9986	1.0077	5.3466
	Heavy	0.7338	0.9858	0.9969	1.0045	2.2081
30	**Small**	0.8607	0.9825	0.9941	1.0066	1.0231
	Large	0.2569	0.9920	1.0000	1.0124	4.3745
	Heavy	0.6638	0.9895	0.9984	1.0088	3.0772
25	**Small**	0.9503	0.9878	1.0006	1.0072	1.0216
	Large	0.5776	0.9931	0.9989	1.0037	5.1357
	Heavy	0.8067	0.9913	0.9972	1.0027	1.3087
20	**Small**	0.9068	0.9874	0.9976	1.0011	1.0290
	Large	0.4532	0.9962	1.0000	1.0052	4.2523
	Heavy	0.7223	0.9956	0.9994	1.0044	1.3115
15	**Small**	0.9664	0.9956	0.9997	1.0012	1.0134
	Large	0.4981	0.9986	1.0000	1.0021	1.4764
	Heavy	0.7958	0.9976	0.9998	1.0030	3.0519
5	**Small**	0.9996	0.9999	1.0000	1.0007	1.0092
	Large	0.7946	0.9998	1.0000	1.0001	1.1455
	Heavy	0.9682	0.9997	1.0000	1.0006	1.1586
1 (Unnormalized)	**Small**	24.0748	52.5366	78.7659	94.8760	133.5611
	Large	8.6747	97.4668	140.7255	173.8522	540.0005
	Heavy	34.6171	176.9382	277.0461	342.4944	607.2246

TABLE 50. DATA SUMMARY FOR FIGURE 41.

		RMS_Z^{norm}				
Height Resolution (m)	**Analysis Method**	**Lower Whisker**	**Lower Quartile**	**Median**	**Upper Quartile**	**Upper Whisker**
100	**TDP**	0.9821	0.9990	1.0036	1.0086	1.0195
	APA	0.9821	0.9974	0.9997	1.0029	1.0196
40	**TDP**	0.9817	0.9974	1.0003	1.0034	1.0193
	APA	0.9818	0.9990	1.0000	1.0009	1.0184
30	**TDP**	0.9817	0.9976	1.0002	1.0027	1.0195
	APA	0.9842	0.9992	1.0000	1.0004	1.0195
25	**TDP**	0.9818	0.9978	1.0002	1.0030	1.0196
	APA	0.9819	0.9993	0.9999	1.0003	1.0191
20	**TDP**	0.9818	0.9977	1.0000	1.0027	1.0194
	APA	0.9831	0.9994	1.0000	1.0002	1.0188
15	**TDP**	0.9819	0.9982	1.0001	1.0028	1.0196
	APA	0.9819	0.9996	1.0000	1.0001	1.0190
5	**TDP**	0.9825	0.9991	0.9999	1.0003	1.0187
	APA	0.9824	0.9999	1.0000	1.0000	1.0147
1 (Unnormalized)	**TDP**	25.4849	118.1129	170.1740	209.9115	607.2246
	APA	8.6747	97.9444	135.8603	164.6081	535.4947

TABLE 51. DATA SUMMARY FOR FIGURE 42.

		RMS_Z^{norm}				
Height Resolution (m)	**Location**	**Lower Whisker**	**Lower Quartile**	**Median**	**Upper Quartile**	**Upper Whisker**
100	**MEM**	0.9821	0.9977	1.0005	1.0055	1.0193
	DFW	0.9821	0.9975	1.0021	1.0112	1.0196
40	**MEM**	0.9818	0.9984	1.0000	1.0012	1.0184
	DFW	0.9817	0.9982	1.0003	1.0037	1.0193
30	**MEM**	0.9818	0.9989	1.0000	1.0010	1.0195
	DFW	0.9817	0.9982	1.0000	1.0019	1.0192
25	**MEM**	0.9819	0.9989	1.0000	1.0008	1.0191
	DFW	0.9818	0.9989	1.0000	1.0011	1.0196
20	**MEM**	0.9825	0.9991	1.0000	1.0006	1.0184
	DFW	0.9818	0.9988	1.0000	1.0011	1.0194
15	**MEM**	0.9819	0.9993	1.0000	1.0005	1.0190
	DFW	0.9819	0.9987	1.0000	1.0017	1.0196
5	**MEM**	0.9824	0.9999	1.0000	1.0001	1.0187
	DFW	0.9825	0.9994	0.9999	1.0000	1.0143
1 (Unnormalized)	**MEM**	24.0748	140.6648	164.6570	210.1370	413.5125
	DFW	8.6747	78.4228	114.0076	170.5730	607.2246

TABLE 52. DATA SUMMARY FOR FIGURE 43.

		RMS_Z^{norm}				
Height Resolution (m)	**Aircraft Weight Class**	**Lower Whisker**	**Lower Quartile**	**Median**	**Upper Quartile**	**Upper Whisker**
100	**Small**	0.9822	0.9990	1.0012	1.0043	1.0136
	Large	0.9821	0.9977	1.0007	1.0064	1.0196
	Heavy	0.9821	0.9982	1.0010	1.0089	1.0195
40	**Small**	0.9981	0.9992	0.9999	1.0024	1.0080
	Large	0.9817	0.9981	1.0000	1.0018	1.0193
	Heavy	0.9828	0.9996	1.0005	1.0024	1.0190
30	**Small**	0.9956	0.9983	1.0001	1.0025	1.0057
	Large	0.9817	0.9986	1.0000	1.0011	1.0195
	Heavy	0.9820	0.9993	1.0002	1.0017	1.0189
25	**Small**	0.9976	0.9999	1.0000	1.0034	1.0064
	Large	0.9818	0.9988	0.9999	1.0008	1.0196
	Heavy	0.9833	0.9991	1.0000	1.0015	1.0194
20	**Small**	0.9955	0.9995	1.0003	1.0041	1.0065
	Large	0.9818	0.9989	0.9999	1.0007	1.0194
	Heavy	0.9836	0.9996	1.0000	1.0010	1.0193
15	**Small**	0.9985	0.9989	0.9999	1.0015	1.0038
	Large	0.9819	0.9990	1.0000	1.0006	1.0196
	Heavy	0.9823	0.9995	1.0000	1.0009	1.0192
5	**Small**	0.9976	0.9995	0.9999	1.0000	1.0001
	Large	0.9824	0.9996	1.0000	1.0000	1.0187
	Heavy	0.9851	0.9994	1.0000	1.0001	1.0126
1 (Unnormalized)	**Small**	24.0748	54.5802	85.7150	105.1406	133.5611
	Large	8.6747	103.0420	143.7410	177.2162	540.0005
	Heavy	34.6171	176.7393	276.5097	342.0049	607.2246

APPENDIX C: MATLAB CODES FOR SCORING METHODOLOGIES

The MATLAB codes of the three scoring methodologies discussed previously are represented in five script files:

- format_data.m – Converts all data from Results_All xls-file into a mat-file used by the scoring methodologies. Calculates normalized residuals and stores the result in the same mat-file. (See comments in this code for format of resulting mat-file).

- plot_residuals.m – Plots the mean normalized residuals for all aircraft types, locations, and wake models, as seen in this report.

- plot_correlation_coeff.m – Plots the correlation coefficient distribution among all aircraft types, locations, and wake models, as seen in this report.

- plot_cross_correlation.m – Plots the mean cross-correlation vs. lag time among all aircraft types, locations, and wake models, as seen in this report. Can also plot the actual cross-correlation values before the mean is taken.

- plot_corridor_exit.m – Plots the measured vs. predicted corridor exit time and calculates the line of best fit. User can change the dimensions of the specified corridor.

All the scripts are well commented for ease of use.

format_data.m

```
clear all;
close all; clc;

%%% ----------------------------------------------------------------------
% This script contains the code necessary to convert the data in
% Results_All_20100507.xls to a format usable by the scoring methodologies.
% The xls-file was included in the CD given to the NIA and FAA in August 2010.
% The file ALL_resid.mat contains the output from this script and is
% already included in the /MATLAB_Codes/ directory.

% The output file ALL_resid.mat has two variables: data and resid:

% The data variable is a cell array formated as follows:
% column 1:  TRAJEC file name from APA Suite
% column 2:  Wake model (APA/TDP)
% column 3:  Location (DFW/MEM)
% column 4:  Aircraft weight class (Small/Large/Heavy)
% column 5:  Aircraft wing span (m)
% column 6:  Initial vortex circulation strength (m2/s)
% column 7:  Matrix of measured data interpolated to natural number scale:
%            column 1: time
%            column 2: circulation
%            column 3: y-position
%            column 4: z-position
% column 8:  Matrix of predicted data interpolated to natural number scale:
%            column 1: time
%            column 2: circulation
%            column 3: y-position
%            column 4: z-position
% column 9:  Matrix of predicted data at 1m height resolution
% column 10: Matrix of predicted data at 5m height resolution
% column 11: Matrix of predicted data at 15m height resolution
% column 12: Matrix of predicted data at 20m height resolution
% column 13: Matrix of predicted data at 25m height resolution
% column 14: Matrix of predicted data at 30m height resolution
% column 15: Matrix of predicted data at 40m height resolution
% column 16: Matrix of predicted data at 100m height resolution

% The resid variable is a cell array formated as follows:
% column 1: Matrix of normalized residuals for original predicted data:
%           column 1: time
%           column 2: circulation
%           column 3: y-position
%           column 4: z-position
% column 2: Matrix of normalized residuals for 1m height resolution
% column 3: Matrix of normalized residuals for 5m height resolution
% column 4: Matrix of normalized residuals for 15m height resolution
% column 5: Matrix of normalized residuals for 20m height resolution
% column 6: Matrix of normalized residuals for 25m height resolution
% column 7: Matrix of normalized residuals for 30m height resolution
% column 8: Matrix of normalized residuals for 40m height resolution
% column 9: Matrix of normalized residuals for 100m height resolution

%%% ----------------------------------------------------------------------

% % Run this section separately for each sheet in the xls-file --------------
% file = 'Results_All_20100507.xls';
% sheet = 'MEM_TDP3'; % change this to match the sheet name being run
% range = 'A4:BC36181'; % change the bounds to match those found in the sheet
%
% [numeric, txt, raw] = xlsread( file, sheet, range );
%
% save MEM_TDP3_raw raw % change this name to represent the sheet name as shown
% % ------------------------------------------------------------------------
```

```
% raw{1,1} = 'TRAJEC_991110_000000_DFWGT2678_APA.dat';

[rows, cols] = size(raw);

% Run the code below once all sheets have been extracted to combine all
% data -----------------------------------------------------------------
% % Combine files from all locations and methods ---------------------------
% load DFW_APA_raw;  DFW_APA  = raw;
% load DFW_APA2_raw; DFW_APA2 = raw;
% raw = [DFW_APA; DFW_APA2];
% save DFW_APA_all raw
%
% load DFW_TDP_raw;  DFW_TDP  = raw;
% load DFW_TDP2_raw; DFW_TDP2 = raw;
% raw = [DFW_TDP; DFW_TDP2];
% save DFW_TDP_all raw
%
% raw = [DFW_APA; DFW_APA2; DFW_TDP; DFW_TDP2];
% save DFW_all raw
%
% load MEM_APA_raw;  MEM_APA  = raw;
% load MEM_APA2_raw; MEM_APA2 = raw;
% raw = [MEM_APA; MEM_APA2];
% save MEM_APA_all raw
%
% load MEM_TDP_raw;  MEM_TDP  = raw;
% load MEM_TDP2_raw; MEM_TDP2 = raw;
% load MEM_TDP3_raw; MEM_TDP3 = raw;
% raw = [MEM_TDP; MEM_TDP2; MEM_TDP3];
% save MEM_TDP_all raw
%
% raw = [MEM_APA; MEM_APA2; MEM_TDP; MEM_TDP2; MEM_TDP3];
% save MEM_all raw

% ---------------------------------------------------------------------

%%% Load File
% load MEM_APA_all;

[rows, cols] = size(raw);

% find unique filenames
clear curfile predfiles ipredfiles filecount;

curfile = raw{1,1};
predfiles{1,1} = curfile;
ipredfiles(1,1) = 1;
filecount = 2;

for i = 2:rows

    if ~strcmp( raw{i,1}, curfile )

        predfiles{ filecount, 1 } = raw{i,1};
        ipredfiles( filecount, 1 ) = i;

        curfile = raw{i,1};
        filecount = filecount + 1;

    end

end

% get data
clear data;
data = cell( length(predfiles), 16 );
for i = 1:length(ipredfiles)

    if i ~= length(ipredfiles)
```

```
        iend = ipredfiles(i+1,1) - 1;
    else
        iend = rows;
    end

    % predfile name
    data{ i, 1 } = raw{ ipredfiles(i,1), 1 };

    % analysis type
    data{ i, 2 } = raw{ ipredfiles(i,1), 3 };

    % location
    data{ i, 3 } = raw{ ipredfiles(i,1), 5 };

    % aircraft class
    data{ i, 4 } = raw{ ipredfiles(i,1), 7 };

    % aircraft wing span
    data{ i, 5 } = raw{ ipredfiles(i,1), 9 };

    % intitial circulation
    data{ i, 6 } = raw{ ipredfiles(i,1), 10 };

    % measured data
    data{ i, 7 } = cell2mat( raw( ipredfiles(i,1):iend, 14:18 ) );

    % predicted data - original
    data{ i, 8 } = cell2mat( raw( ipredfiles(i,1):iend, 20:23 ) );

    % predicted data - 1m
    data{ i, 9 } = cell2mat( raw( ipredfiles(i,1):iend, 24:27 ) );

    % predicted data - 5m
    data{ i, 10 } = cell2mat( raw( ipredfiles(i,1):iend, 28:31 ) );

    % predicted data - 15m
    data{ i, 11 } = cell2mat( raw( ipredfiles(i,1):iend, 32:35 ) );

    % predicted data - 20m
    data{ i, 12 } = cell2mat( raw( ipredfiles(i,1):iend, 36:39 ) );

    % predicted data - 25m
    data{ i, 13 } = cell2mat( raw( ipredfiles(i,1):iend, 40:43 ) );

    % predicted data - 30m
    data{ i, 14 } = cell2mat( raw( ipredfiles(i,1):iend, 44:47 ) );

    % predicted data - 40m
    data{ i, 15 } = cell2mat( raw( ipredfiles(i,1):iend, 48:51 ) );

    % predicted data - 100m
    data{ i, 16 } = cell2mat( raw( ipredfiles(i,1):iend, 52:55 ) );

end

% remove NaNs from measured data and predicted data
for i = 1:length(data)

    data{i,7}(isnan( data{i,7}(:,1) ), :) = [];
    data{i,8}(isnan( data{i,8}(:,1) ), :) = [];
    data{i,9}(isnan( data{i,9}(:,1) ), :) = [];
    data{i,10}(isnan( data{i,10}(:,1) ), :) = [];
    data{i,11}(isnan( data{i,11}(:,1) ), :) = [];
    data{i,12}(isnan( data{i,12}(:,1) ), :) = [];
    data{i,13}(isnan( data{i,13}(:,1) ), :) = [];
    data{i,14}(isnan( data{i,14}(:,1) ), :) = [];
    data{i,15}(isnan( data{i,15}(:,1) ), :) = [];
    data{i,16}(isnan( data{i,16}(:,1) ), :) = [];

end
```

```
% remove data with no measured data or only one data point or N/A for
% initial circ.
for i = length(data):-1:1
    [rowstest, colstest] = size( data{i,7} );
    if isempty( data{i,7} ) || rowstest == 1 || strcmp( data{i,6}, 'N/A' )
        data(i,:) = [];
    end
end
data(108,:) = [];   % MEM APA
% data(502:504,:) = [];   % MEM TDP

% calculate residuals
resid = cell( length(data), 9 );
for i = 1:length(data)

    % offset measured data
    data{i,7}(:,1) = data{i,7}(:,1) - data{i,7}(:,5);

    % interpolate measured data
    inttime = ( ceil( data{i,7}(1,1) ) : 1 : floor( max(data{i,7}(:,1)) ) )';

    if isequal( inttime, data{i,7}(:,1) )
        intdata = data{i,7}(:,2:4);
    else
        % remove trailing zeros
        if length( data{i,7}(:,1) ) > 1
            while data{i,7}(end,1) <= data{i,7}(end-1,1)
                data{i,7}(end,:) = [];
            end
        end
        intdata = interp1( data{i,7}(:,1), data{i,7}(:,2:4), inttime, 'spline' );
    end

    data{i,7} = [inttime intdata];

    % process predicted data
    for j = 8:16

        if ~isempty( data{i,j} )
            % adjust Z-data from 100m
            data{i,j}(:,4) = data{i,j}(:,4) - (100 - data{i,7}(1,4));

            % interpolate predicted data
            inttime = ( ceil( data{i,j}(1,1) ) : 1 : floor( data{i,j}(end,1) ) )';

            if isequal( inttime, data{i,j}(:,1) )
                intdata = data{i,j}(:,2:end);
            else
                intdata = interp1( data{i,j}(:,1), data{i,j}(:,2:end), inttime,
'spline' );
            end

            data{i,j} = [inttime intdata];

            % calculate residuals
            mintime = max( min(data{i,7}(:,1)), min(data{i,j}(:,1)) );
            maxtime = min( max(data{i,7}(:,1)), max(data{i,j}(:,1)) );

            ismeas = find(data{i,7}(:,1) == mintime);
            ispred = find(data{i,j}(:,1) == mintime);

            iemeas = find(data{i,7}(:,1) == maxtime);
            iepred = find(data{i,j}(:,1) == maxtime);

            resid{i,j-7}(:,1) = data{i,7}(ismeas:iemeas,1);
            resid{i,j-7}(:,2) = ( abs( data{i,7}(ismeas:iemeas,2) ) - abs(
data{i,j}(ispred:iepred,2) ) )/data{i,6};
            resid{i,j-7}(:,3) = ( data{i,7}(ismeas:iemeas,3) -
data{i,j}(ispred:iepred,3) )/data{i,5};
```

```
            resid{i,j-7}(:,4) = ( data{i,7}(ismeas:iemeas,4) -
data{i,j}(ispred:iepred,4) )/data{i,5};
        end

    end

%     clear resid
%     resid(:,1) = data{i,7}(ismeas:iemeas,1);
%     resid(:,2) = ( abs( data{i,7}(ismeas:iemeas,2) ) - abs(
data{i,8}(ispred:iepred,2) ) )/data{i,6};
%     resid(:,3) = ( abs( data{i,7}(ismeas:iemeas,3) ) - abs(
data{i,8}(ispred:iepred,3) ) )/data{i,5};
%     resid(:,4) = ( abs( data{i,7}(ismeas:iemeas,4) ) - abs(
data{i,8}(ispred:iepred,4) ) )/data{i,5};
%
%     data{i,9} = resid;
end


% Save resid and data into same mat-file ----------------------------------
load DFW_APA_resid
DFW_APA_data  = data;
DFW_APA_resid = resid;

load DFW_TDP_resid
DFW_TDP_data  = data;
DFW_TDP_resid = resid;

load MEM_APA_resid
MEM_APA_data  = data;
MEM_APA_resid = resid;

load MEM_TDP_resid
MEM_TDP_data  = data;
MEM_TDP_resid = resid;

data = [DFW_APA_data
        DFW_TDP_data
        MEM_APA_data
        MEM_TDP_data];
resid = [DFW_APA_resid
         DFW_TDP_resid
         MEM_APA_resid
         MEM_TDP_resid];

save ALL_resid data resid

load ALL_resid
```

PLOT_RESIDUALS.m

```
% Plot Normalized Residuals
% This script plots the averaged normalized residuals vs. time. The
% residuals are loaded directly from the ALL_resid file, and are
% precalcualted.
% All plots are split up by location, aircraft type, and wake model.
clear all; close all; clc;

%%%------------------------ BEGIN USER INPUTS --------------------------%%%

% Load data file
load ALL_resid

speed = 180;    % following aircraft speed relative to deposited vortex (kts)


%%%------------------------ END OF USER INPUTS -------------------------%%%

speedms = speed/1.94384449;
xstring = ['Follow Distance @ ', num2str(speed), 'kts (m)'];

small_dfw_c_apa(:,1) = (0:200)';
small_dfw_c_tdp(:,1) = (0:200)';
small_dfw_y_apa(:,1) = (0:200)';
small_dfw_y_tdp(:,1) = (0:200)';
small_dfw_z_apa(:,1) = (0:200)';
small_dfw_z_tdp(:,1) = (0:200)';
small_mem_c_apa(:,1) = (0:200)';
small_mem_c_tdp(:,1) = (0:200)';
small_mem_y_apa(:,1) = (0:200)';
small_mem_y_tdp(:,1) = (0:200)';
small_mem_z_apa(:,1) = (0:200)';
small_mem_z_tdp(:,1) = (0:200)';

large_dfw_c_apa(:,1) = (0:200)';
large_dfw_c_tdp(:,1) = (0:200)';
large_dfw_y_apa(:,1) = (0:200)';
large_dfw_y_tdp(:,1) = (0:200)';
large_dfw_z_apa(:,1) = (0:200)';
large_dfw_z_tdp(:,1) = (0:200)';
large_mem_c_apa(:,1) = (0:200)';
large_mem_c_tdp(:,1) = (0:200)';
large_mem_y_apa(:,1) = (0:200)';
large_mem_y_tdp(:,1) = (0:200)';
large_mem_z_apa(:,1) = (0:200)';
large_mem_z_tdp(:,1) = (0:200)';

heavy_dfw_c_apa(:,1) = (0:200)';
heavy_dfw_c_tdp(:,1) = (0:200)';
heavy_dfw_y_apa(:,1) = (0:200)';
heavy_dfw_y_tdp(:,1) = (0:200)';
heavy_dfw_z_apa(:,1) = (0:200)';
heavy_dfw_z_tdp(:,1) = (0:200)';
heavy_mem_c_apa(:,1) = (0:200)';
heavy_mem_c_tdp(:,1) = (0:200)';
heavy_mem_y_apa(:,1) = (0:200)';
heavy_mem_y_tdp(:,1) = (0:200)';
heavy_mem_z_apa(:,1) = (0:200)';
heavy_mem_z_tdp(:,1) = (0:200)';


% get residuals for each configuration
for i = 1:length(data)

    % DFW - small - circulation
```

```
    if ~isempty( resid{i,2} ) && strcmp( data{i,4}, 'Small' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'APA' )

        foo1 = small_dfw_c_apa;
        foot = resid{i,2}(1,1);
        foo2 = resid{i,2}(:,2);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(201-ind-rows,1)] ];

        small_dfw_c_apa = foo1;

    end

    if ~isempty( resid{i,2} ) && strcmp( data{i,4}, 'Small' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = small_dfw_c_tdp;
        foot = resid{i,2}(1,1);
        foo2 = resid{i,2}(:,2);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(201-ind-rows,1)] ];

        small_dfw_c_tdp = foo1;

    end

    % DFW - small - y
    if ~isempty( resid{i,2} ) && strcmp( data{i,4}, 'Small' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'APA' )

        foo1 = small_dfw_y_apa;
        foot = resid{i,2}(1,1);
        foo2 = resid{i,2}(:,3);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(201-ind-rows,1)] ];

        small_dfw_y_apa = foo1;

    end

    if ~isempty( resid{i,2} ) && strcmp( data{i,4}, 'Small' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = small_dfw_y_tdp;
        foot = resid{i,2}(1,1);
        foo2 = resid{i,2}(:,3);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(201-ind-rows,1)] ];

        small_dfw_y_tdp = foo1;

    end

    % DFW - small - z
    if ~isempty( resid{i,2} ) && strcmp( data{i,4}, 'Small' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'APA' )

        foo1 = small_dfw_z_apa;
        foot = resid{i,2}(1,1);
        foo2 = resid{i,2}(:,4);

        [rows,cols] = size( foo2 );
```

```
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(201-ind-rows,1)] ];

        small_dfw_z_apa = foo1;

    end

    if ~isempty( resid{i,2} ) && strcmp( data{i,4}, 'Small' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = small_dfw_z_tdp;
        foot = resid{i,2}(1,1);
        foo2 = resid{i,2}(:,4);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(201-ind-rows,1)] ];

        small_dfw_z_tdp = foo1;

    end

    % MEM - small - circulation
    if ~isempty( resid{i,2} ) && strcmp( data{i,4}, 'Small' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'APA' )

        foo1 = small_mem_c_apa;
        foot = resid{i,2}(1,1);
        foo2 = resid{i,2}(:,2);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(201-ind-rows,1)] ];

        small_mem_c_apa = foo1;

    end

    if ~isempty( resid{i,2} ) && strcmp( data{i,4}, 'Small' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = small_mem_c_tdp;
        foot = resid{i,2}(1,1);
        foo2 = resid{i,2}(:,2);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(201-ind-rows,1)] ];

        small_mem_c_tdp = foo1;

    end

    % MEM residuals - small - y
    if ~isempty( resid{i,2} ) && strcmp( data{i,4}, 'Small' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'APA' )

        foo1 = small_mem_y_apa;
        foot = resid{i,2}(1,1);
        foo2 = resid{i,2}(:,3);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(201-ind-rows,1)] ];

        small_mem_y_apa = foo1;

    end
```

```
    if ~isempty( resid{i,2} ) && strcmp( data{i,4}, 'Small' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = small_mem_y_tdp;
        foot = resid{i,2}(1,1);
        foo2 = resid{i,2}(:,3);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(201-ind-rows,1)] ];

        small_mem_y_tdp = foo1;

    end

    % MEM - small - z
    if ~isempty( resid{i,2} ) && strcmp( data{i,4}, 'Small' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'APA' )

        foo1 = small_mem_z_apa;
        foot = resid{i,2}(1,1);
        foo2 = resid{i,2}(:,4);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(201-ind-rows,1)] ];

        small_mem_z_apa = foo1;

    end

    if ~isempty( resid{i,2} ) && strcmp( data{i,4}, 'Small' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = small_mem_z_tdp;
        foot = resid{i,2}(1,1);
        foo2 = resid{i,2}(:,4);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(201-ind-rows,1)] ];

        small_mem_z_tdp = foo1;

    end

    % DFW - large - circulation
    if ~isempty( resid{i,2} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'APA' )

        foo1 = large_dfw_c_apa;
        foot = resid{i,2}(1,1);
        foo2 = resid{i,2}(:,2);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(201-ind-rows,1)] ];

        large_dfw_c_apa = foo1;

    end

    if ~isempty( resid{i,2} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = large_dfw_c_tdp;
        foot = resid{i,2}(1,1);
        foo2 = resid{i,2}(:,2);

        [rows,cols] = size( foo2 );
```

```
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(201-ind-rows,1)] ];

        large_dfw_c_tdp = foo1;

    end

    % DFW - large - y
    if ~isempty( resid{i,2} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'APA' )

        foo1 = large_dfw_y_apa;
        foot = resid{i,2}(1,1);
        foo2 = resid{i,2}(:,3);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(201-ind-rows,1)] ];

        large_dfw_y_apa = foo1;

    end

    if ~isempty( resid{i,2} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = large_dfw_y_tdp;
        foot = resid{i,2}(1,1);
        foo2 = resid{i,2}(:,3);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(201-ind-rows,1)] ];

        large_dfw_y_tdp = foo1;

    end

    % DFW - large - z
    if ~isempty( resid{i,2} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'APA' )

        foo1 = large_dfw_z_apa;
        foot = resid{i,2}(1,1);
        foo2 = resid{i,2}(:,4);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(201-ind-rows,1)] ];

        large_dfw_z_apa = foo1;

    end

    if ~isempty( resid{i,2} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = large_dfw_z_tdp;
        foot = resid{i,2}(1,1);
        foo2 = resid{i,2}(:,4);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(201-ind-rows,1)] ];

        large_dfw_z_tdp = foo1;

    end

    % MEM - large - circulation
```

```
    if ~isempty( resid{i,2} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3}, 
'MEM' ) && strcmp( data{i,2}, 'APA' )

        foo1 = large_mem_c_apa;
        foot = resid{i,2}(1,1);
        foo2 = resid{i,2}(:,2);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(201-ind-rows,1)] ];

        large_mem_c_apa = foo1;

    end

    if ~isempty( resid{i,2} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3}, 
'MEM' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = large_mem_c_tdp;
        foot = resid{i,2}(1,1);
        foo2 = resid{i,2}(:,2);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(201-ind-rows,1)] ];

        large_mem_c_tdp = foo1;

    end

    % MEM - large - y
    if ~isempty( resid{i,2} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3}, 
'MEM' ) && strcmp( data{i,2}, 'APA' )

        foo1 = large_mem_y_apa;
        foot = resid{i,2}(1,1);
        foo2 = resid{i,2}(:,3);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(201-ind-rows,1)] ];

        large_mem_y_apa = foo1;

    end

    if ~isempty( resid{i,2} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3}, 
'MEM' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = large_mem_y_tdp;
        foot = resid{i,2}(1,1);
        foo2 = resid{i,2}(:,3);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(201-ind-rows,1)] ];

        large_mem_y_tdp = foo1;

    end

    % MEM - large - z
    if ~isempty( resid{i,2} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3}, 
'MEM' ) && strcmp( data{i,2}, 'APA' )

        foo1 = large_mem_z_apa;
        foot = resid{i,2}(1,1);
        foo2 = resid{i,2}(:,4);

        [rows,cols] = size( foo2 );
```

```
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(201-ind-rows,1)] ];

        large_mem_z_apa = foo1;

    end

    if ~isempty( resid{i,2} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = large_mem_z_tdp;
        foot = resid{i,2}(1,1);
        foo2 = resid{i,2}(:,4);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(201-ind-rows,1)] ];

        large_mem_z_tdp = foo1;

    end

    % DFW - heavy - circulation
    if ~isempty( resid{i,2} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'APA' )

        foo1 = heavy_dfw_c_apa;
        foot = resid{i,2}(1,1);
        foo2 = resid{i,2}(:,2);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(201-ind-rows,1)] ];

        heavy_dfw_c_apa = foo1;

    end

    if ~isempty( resid{i,2} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = heavy_dfw_c_tdp;
        foot = resid{i,2}(1,1);
        foo2 = resid{i,2}(:,2);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(201-ind-rows,1)] ];

        heavy_dfw_c_tdp = foo1;

    end

    % DFW - heavy - y
    if ~isempty( resid{i,2} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'APA' )

        foo1 = heavy_dfw_y_apa;
        foot = resid{i,2}(1,1);
        foo2 = resid{i,2}(:,3);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(201-ind-rows,1)] ];

        heavy_dfw_y_apa = foo1;

    end
```

```
    if ~isempty( resid{i,2} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = heavy_dfw_y_tdp;
        foot = resid{i,2}(1,1);
        foo2 = resid{i,2}(:,3);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(201-ind-rows,1)] ];

        heavy_dfw_y_tdp = foo1;

    end

    % DFW - heavy - z
    if ~isempty( resid{i,2} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'APA' )

        foo1 = heavy_dfw_z_apa;
        foot = resid{i,2}(1,1);
        foo2 = resid{i,2}(:,4);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(201-ind-rows,1)] ];

        heavy_dfw_z_apa = foo1;

    end

    if ~isempty( resid{i,2} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = heavy_dfw_z_tdp;
        foot = resid{i,2}(1,1);
        foo2 = resid{i,2}(:,4);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(201-ind-rows,1)] ];

        heavy_dfw_z_tdp = foo1;

    end

    % MEM - heavy - circulation
    if ~isempty( resid{i,2} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'APA' )

        foo1 = heavy_mem_c_apa;
        foot = resid{i,2}(1,1);
        foo2 = resid{i,2}(:,2);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(201-ind-rows,1)] ];

        heavy_mem_c_apa = foo1;

    end

    if ~isempty( resid{i,2} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = heavy_mem_c_tdp;
        foot = resid{i,2}(1,1);
        foo2 = resid{i,2}(:,2);

        [rows,cols] = size( foo2 );
```

```
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(201-ind-rows,1)] ];

        heavy_mem_c_tdp = foo1;

    end

    % MEM - heavy - y
    if ~isempty( resid{i,2} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'APA' )

        foo1 = heavy_mem_y_apa;
        foot = resid{i,2}(1,1);
        foo2 = resid{i,2}(:,3);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(201-ind-rows,1)] ];

        heavy_mem_y_apa = foo1;

    end

    if ~isempty( resid{i,2} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = heavy_mem_y_tdp;
        foot = resid{i,2}(1,1);
        foo2 = resid{i,2}(:,3);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(201-ind-rows,1)] ];

        heavy_mem_y_tdp = foo1;

    end

    % MEM - heavy - z
    if ~isempty( resid{i,2} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'APA' )

        foo1 = heavy_mem_z_apa;
        foot = resid{i,2}(1,1);
        foo2 = resid{i,2}(:,4);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(201-ind-rows,1)] ];

        heavy_mem_z_apa = foo1;

    end

    if ~isempty( resid{i,2} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = heavy_mem_z_tdp;
        foot = resid{i,2}(1,1);
        foo2 = resid{i,2}(:,4);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(201-ind-rows,1)] ];

        heavy_mem_z_tdp = foo1;

    end
end
```

```
% Find Min, Mean, Max
varlabels = { 'small_dfw_c_apa', 'small_dfw_c_tdp', 'small_dfw_y_apa',
'small_dfw_y_tdp', 'small_dfw_z_apa', 'small_dfw_z_tdp', ...
              'small_mem_c_apa', 'small_mem_c_tdp', 'small_mem_y_apa',
'small_mem_y_tdp', 'small_mem_z_apa', 'small_mem_z_tdp', ...
              'large_dfw_c_apa', 'large_dfw_c_tdp', 'large_dfw_y_apa',
'large_dfw_y_tdp', 'large_dfw_z_apa', 'large_dfw_z_tdp', ...
              'large_mem_c_apa', 'large_mem_c_tdp', 'large_mem_y_apa',
'large_mem_y_tdp', 'large_mem_z_apa', 'large_mem_z_tdp', ...
              'heavy_dfw_c_apa', 'heavy_dfw_c_tdp', 'heavy_dfw_y_apa',
'heavy_dfw_y_tdp', 'heavy_dfw_z_apa', 'heavy_dfw_z_tdp', ...
              'heavy_mem_c_apa', 'heavy_mem_c_tdp', 'heavy_mem_y_apa',
'heavy_mem_y_tdp', 'heavy_mem_z_apa', 'heavy_mem_z_tdp' };

for i = 1:length( varlabels )

    eval( ['mmm_', varlabels{i}, '(:,2) = nanmean( ', varlabels{i}, '(:,2:end), 2 );']
);
    eval( ['mmm_', varlabels{i}, '(:,4) =  nanstd( ', varlabels{i}, '(:,2:end), 0, 2
);'] );

    eval( ['mmm_', varlabels{i}, '(:,1) =  mmm_', varlabels{i}, '(:,2) - 3*mmm_',
varlabels{i}, '(:,4);'] );
    eval( ['mmm_', varlabels{i}, '(:,3) =  mmm_', varlabels{i}, '(:,2) + 3*mmm_',
varlabels{i}, '(:,4);'] );

end


% Plot Results
% SMALL ------------------------------------------------------------------
figure( 'Name', 'Small - C' )
subplot( 1, 2, 1 )
plot( (foo1(:,1)-1)*speedms, mmm_small_dfw_c_apa(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_small_dfw_c_tdp(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_small_dfw_c_apa(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_small_dfw_c_apa(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_small_dfw_c_tdp(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_small_dfw_c_tdp(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Normalized Circulation Residual, DFW'} )
axis( [0 2500 -3 3] )

subplot( 1, 2, 2 )
plot( (foo1(:,1)-1)*speedms, mmm_small_mem_c_apa(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_small_mem_c_tdp(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_small_mem_c_apa(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_small_mem_c_apa(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_small_mem_c_tdp(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_small_mem_c_tdp(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Normalized Circulation Residual, MEM'} )
axis( [0 2500 -3 3] )


figure( 'Name', 'Small - Y' )
subplot( 1, 2, 1 )
```

```matlab
plot( (foo1(:,1)-1)*speedms, mmm_small_dfw_y_apa(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_small_dfw_y_tdp(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_small_dfw_y_apa(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_small_dfw_y_apa(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_small_dfw_y_tdp(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_small_dfw_y_tdp(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Normalized Y-Pos. Residual, DFW'} )
axis( [0 2500 -8 10] )

subplot( 1, 2, 2 )
plot( (foo1(:,1)-1)*speedms, mmm_small_mem_y_apa(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_small_mem_y_tdp(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_small_mem_y_apa(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_small_mem_y_apa(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_small_mem_y_tdp(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_small_mem_y_tdp(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Normalized Y-Pos. Residual, MEM'} )
axis( [0 2500 -8 10] )


figure( 'Name', 'Small - Z' )
subplot( 1, 2, 1 )
plot( (foo1(:,1)-1)*speedms, mmm_small_dfw_z_apa(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_small_dfw_z_tdp(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_small_dfw_z_apa(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_small_dfw_z_apa(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_small_dfw_z_tdp(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_small_dfw_z_tdp(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Normalized Z-Pos. Residual, DFW'} )
axis( [0 2500 -2 5] )

subplot( 1, 2, 2 )
plot( (foo1(:,1)-1)*speedms, mmm_small_mem_z_apa(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_small_mem_z_tdp(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_small_mem_z_apa(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_small_mem_z_apa(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_small_mem_z_tdp(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_small_mem_z_tdp(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Normalized Z-Pos. Residual, MEM'} )
axis( [0 2500 -2 5] )
```

```matlab
% LARGE -----------------------------------------------------------------
figure( 'Name', 'large - C' )
subplot( 1, 2, 1 )
plot( (foo1(:,1)-1)*speedms, mmm_large_dfw_c_apa(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_large_dfw_c_tdp(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_large_dfw_c_apa(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_large_dfw_c_apa(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_large_dfw_c_tdp(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_large_dfw_c_tdp(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Normalized Circulation Residual, DFW'} )
axis( [0 13000 -10 6] )

subplot( 1, 2, 2 )
plot( (foo1(:,1)-1)*speedms, mmm_large_mem_c_apa(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_large_mem_c_tdp(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_large_mem_c_apa(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_large_mem_c_apa(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_large_mem_c_tdp(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_large_mem_c_tdp(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Normalized Circulation Residual, MEM'} )
axis( [0 13000 -10 6] )


figure( 'Name', 'large - Y' )
subplot( 1, 2, 1 )
plot( (foo1(:,1)-1)*speedms, mmm_large_dfw_y_apa(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_large_dfw_y_tdp(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_large_dfw_y_apa(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_large_dfw_y_apa(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_large_dfw_y_tdp(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_large_dfw_y_tdp(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Normalized Y-Pos. Residual, DFW'} )
axis( [0 13000 -20 20] )

subplot( 1, 2, 2 )
plot( (foo1(:,1)-1)*speedms, mmm_large_mem_y_apa(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_large_mem_y_tdp(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_large_mem_y_apa(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_large_mem_y_apa(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_large_mem_y_tdp(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_large_mem_y_tdp(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Normalized Y-Pos. Residual, MEM'} )
```

```
axis( [0 13000 -20 20] )


figure( 'Name', 'large - Z' )
subplot( 1, 2, 1 )
plot( (foo1(:,1)-1)*speedms, mmm_large_dfw_z_apa(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_large_dfw_z_tdp(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_large_dfw_z_apa(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_large_dfw_z_apa(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_large_dfw_z_tdp(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_large_dfw_z_tdp(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Normalized Z-Pos. Residual, DFW'} )
axis( [0 13000 -12 15] )

subplot( 1, 2, 2 )
plot( (foo1(:,1)-1)*speedms, mmm_large_mem_z_apa(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_large_mem_z_tdp(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_large_mem_z_apa(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_large_mem_z_apa(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_large_mem_z_tdp(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_large_mem_z_tdp(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Normalized Z-Pos. Residual, MEM'} )
axis( [0 13000 -12 15] )


% HEAVY -------------------------------------------------------------
figure( 'Name', 'heavy - C' )
subplot( 1, 2, 1 )
plot( (foo1(:,1)-1)*speedms, mmm_heavy_dfw_c_apa(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_dfw_c_tdp(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_heavy_dfw_c_apa(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_dfw_c_apa(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_dfw_c_tdp(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_dfw_c_tdp(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Normalized Circulation Residual, DFW'} )
axis( [0 13000 -10 8] )

subplot( 1, 2, 2 )
plot( (foo1(:,1)-1)*speedms, mmm_heavy_mem_c_apa(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_mem_c_tdp(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_heavy_mem_c_apa(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_mem_c_apa(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_mem_c_tdp(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_mem_c_tdp(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
```

```
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Normalized Circulation Residual, MEM'} )
axis( [0 13000 -10 8] )


figure( 'Name', 'heavy - Y' )
subplot( 1, 2, 1 )
plot( (foo1(:,1)-1)*speedms, mmm_heavy_dfw_y_apa(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_dfw_y_tdp(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_heavy_dfw_y_apa(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_dfw_y_apa(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_dfw_y_tdp(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_dfw_y_tdp(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Normalized Y-Pos. Residual, DFW'} )
axis( [0 13000 -8 10] )

subplot( 1, 2, 2 )
plot( (foo1(:,1)-1)*speedms, mmm_heavy_mem_y_apa(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_mem_y_tdp(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_heavy_mem_y_apa(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_mem_y_apa(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_mem_y_tdp(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_mem_y_tdp(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Normalized Y-Pos. Residual, MEM'} )
axis( [0 13000 -8 10] )


figure( 'Name', 'heavy - Z' )
subplot( 1, 2, 1 )
plot( (foo1(:,1)-1)*speedms, mmm_heavy_dfw_z_apa(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_dfw_z_tdp(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_heavy_dfw_z_apa(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_dfw_z_apa(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_dfw_z_tdp(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_dfw_z_tdp(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Normalized Z-Pos. Residual, DFW'} )
axis( [0 13000 -6 3] )

subplot( 1, 2, 2 )
plot( (foo1(:,1)-1)*speedms, mmm_heavy_mem_z_apa(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_mem_z_tdp(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_heavy_mem_z_apa(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_mem_z_apa(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_mem_z_tdp(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_mem_z_tdp(:,3), 'b^-', ...
```

```
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Normalized Z-Pos. Residual, MEM'} )
axis( [0 13000 -6 3] )
```

CORRIDOR EXIT FUNCTION

plot_corridor_exit.m

```
% Corridor Exit Method
% This script plots the measured corridor exit time vs. the predicted
% corridor exit time for a user specified corridor
% All plots are split up by location, aircraft type, and wake model.
clear all; close all; clc;

%%%------------------------ BEGIN USER INPUTS --------------------------%%%

% Load data file
load ALL_resid

% Define corridor dimensions
ymax_ft = 537;     % corridor height (ft)
zmax_ft = 114;      % corridor width (ft)


%%%------------------------ END OF USER INPUTS ------------------------%%%

% Get points where data goes over corridor
ymax = ymax_ft*0.3048;
zmax = zmax_ft*0.3048;

imax = length(data);

texit_meas = zeros( imax, 1 );
texit_pred = zeros( imax, 1 );

for i = 1:imax

    wspan = data{i,5}; % wing span
    [rows_meas, cols_meas] = size( data{i,7} );
    [rows_pred, cols_pred] = size( data{i,8} );

    % get exit time for measured data
    if rows_meas > 1    % if measured data exists

        for j = 1:rows_meas

            ypos_meas = data{i,7}(j,3); % measured y-pos
            zpos_meas = data{i,7}(j,4); % measured z-pos

            texit_meas(i,1) = data{i,7}(j,1); % time of vortex exit

            % check if vortex has left corridor
            if abs( ypos_meas ) > 0.5*ymax || abs( zpos_meas - data{i,7}(1,4) ) >
0.5*zmax
                break;
            end

        end

    else % if no measured data exists

        texit_meas(i,1) = NaN; % time of vortex exit

    end

    % get exit time for predicted data
    if rows_pred > 1

        for j = 1:rows_pred

            ypos_pred = data{i,8}(j,3); % predicted y-pos
```

```
            zpos_pred = data{i,8}(j,4); % predicted z-pos

            texit_pred(i,1) = data{i,8}(j,1); % time of vortex exit

            % check if vortex has left corridor
            if abs( ypos_pred ) > 0.5*ymax || abs( zpos_meas - data{i,8}(1,4) ) >
0.5*zmax
                break;
            end

        end

        % if exit time is equal to final time, check for vortex death
        if texit_pred(i,1) == data{i,8}(rows_pred,1)

            for j = rows_pred:-1:1
                if j == 1
                    break;
                elseif data{i,8}(j,2) ~= data{i,8}(j-1,2)
                    texit_pred(i,1) = data{i,8}(j-1,1);
                    break;
                end
            end

        end

    else

        texit_pred(i,1) = NaN; % time of vortex exit

    end

    % adjust all zeros to NaN
    if texit_pred(i,1) == 0
        texit_pred(i,1) = NaN;
    end

end


% Count all rows with different parameters
count_DFW_APA_Small = 0;
count_DFW_APA_Large = 0;
count_DFW_APA_Heavy = 0;
count_DFW_TDP_Small = 0;
count_DFW_TDP_Large = 0;
count_DFW_TDP_Heavy = 0;
count_MEM_APA_Small = 0;
count_MEM_APA_Large = 0;
count_MEM_APA_Heavy = 0;
count_MEM_TDP_Small = 0;
count_MEM_TDP_Large = 0;
count_MEM_TDP_Heavy = 0;

for i = 1:imax

    if strcmp( data{i,3}, 'DFW' )
        if strcmp( data{i,2}, 'APA' )
            if strcmp( data{i,4}, 'Small' )
                count_DFW_APA_Small = count_DFW_APA_Small + 1;
            elseif strcmp( data{i,4}, 'Large' )
                count_DFW_APA_Large = count_DFW_APA_Large + 1;
            elseif strcmp( data{i,4}, 'Heavy' )
                count_DFW_APA_Heavy = count_DFW_APA_Heavy + 1;
            end
        elseif strcmp( data{i,2}, 'TDP' )
            if strcmp( data{i,4}, 'Small' )
                count_DFW_TDP_Small = count_DFW_TDP_Small + 1;
            elseif strcmp( data{i,4}, 'Large' )
                count_DFW_TDP_Large = count_DFW_TDP_Large + 1;
```

```
            elseif strcmp( data{i,4}, 'Heavy' )
                count_DFW_TDP_Heavy = count_DFW_TDP_Heavy + 1;
            end
        end
    elseif strcmp( data{i,3}, 'MEM' )
        if strcmp( data{i,2}, 'APA' )
            if strcmp( data{i,4}, 'Small' )
                count_MEM_APA_Small = count_MEM_APA_Small + 1;
            elseif strcmp( data{i,4}, 'Large' )
                count_MEM_APA_Large = count_MEM_APA_Large + 1;
            elseif strcmp( data{i,4}, 'Heavy' )
                count_MEM_APA_Heavy = count_MEM_APA_Heavy + 1;
            end
        elseif strcmp( data{i,2}, 'TDP' )
            if strcmp( data{i,4}, 'Small' )
                count_MEM_TDP_Small = count_MEM_TDP_Small + 1;
            elseif strcmp( data{i,4}, 'Large' )
                count_MEM_TDP_Large = count_MEM_TDP_Large + 1;
            elseif strcmp( data{i,4}, 'Heavy' )
                count_MEM_TDP_Heavy = count_MEM_TDP_Heavy + 1;
            end
        end
    end

end

% figure;
% plot( texit_pred, texit_meas, 'k.', 'linewidth', 2 )
% xlabel( 'Predicted Exit Time (s)' )
% ylabel( 'Measured Exit Time (s)' )
% grid on

% Separate results by categories
texit_pred_DFW_APA_Small = [];
texit_pred_DFW_APA_Large = [];
texit_pred_DFW_APA_Heavy = [];
texit_pred_DFW_TDP_Small = [];
texit_pred_DFW_TDP_Large = [];
texit_pred_DFW_TDP_Heavy = [];
texit_meas_DFW_APA_Small = [];
texit_meas_DFW_APA_Large = [];
texit_meas_DFW_APA_Heavy = [];
texit_meas_DFW_TDP_Small = [];
texit_meas_DFW_TDP_Large = [];
texit_meas_DFW_TDP_Heavy = [];

texit_pred_MEM_APA_Small = [];
texit_pred_MEM_APA_Large = [];
texit_pred_MEM_APA_Heavy = [];
texit_pred_MEM_TDP_Small = [];
texit_pred_MEM_TDP_Large = [];
texit_pred_MEM_TDP_Heavy = [];
texit_meas_MEM_APA_Small = [];
texit_meas_MEM_APA_Large = [];
texit_meas_MEM_APA_Heavy = [];
texit_meas_MEM_TDP_Small = [];
texit_meas_MEM_TDP_Large = [];
texit_meas_MEM_TDP_Heavy = [];

%%% DFW
% Get exit times - small
for i = 1:imax

    if strcmp( data{i,4}, 'Small' ) && strcmp( data{i,3}, 'DFW' ) && strcmp(
data{i,2}, 'APA' )
        texit_pred_DFW_APA_Small = [texit_pred_DFW_APA_Small; texit_pred(i,1)];
        texit_meas_DFW_APA_Small = [texit_meas_DFW_APA_Small; texit_meas(i,1)];
    end
```

```
    if strcmp( data{i,4}, 'Small' ) && strcmp( data{i,3}, 'DFW' ) && strcmp(
data{i,2}, 'TDP' )
        texit_pred_DFW_TDP_Small = [texit_pred_DFW_TDP_Small; texit_pred(i,1)];
        texit_meas_DFW_TDP_Small = [texit_meas_DFW_TDP_Small; texit_meas(i,1)];
    end

end

% Get exit times - large
for i = 1:imax

    if strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3}, 'DFW' ) && strcmp(
data{i,2}, 'APA' )
        texit_pred_DFW_APA_Large = [texit_pred_DFW_APA_Large; texit_pred(i,1)];
        texit_meas_DFW_APA_Large = [texit_meas_DFW_APA_Large; texit_meas(i,1)];
    end

    if strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3}, 'DFW' ) && strcmp(
data{i,2}, 'TDP' )
        texit_pred_DFW_TDP_Large = [texit_pred_DFW_TDP_Large; texit_pred(i,1)];
        texit_meas_DFW_TDP_Large = [texit_meas_DFW_TDP_Large; texit_meas(i,1)];
    end

end

% Get exit times - heavy
for i = 1:imax

    if strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3}, 'DFW' ) && strcmp(
data{i,2}, 'APA' )
        texit_pred_DFW_APA_Heavy = [texit_pred_DFW_APA_Heavy; texit_pred(i,1)];
        texit_meas_DFW_APA_Heavy = [texit_meas_DFW_APA_Heavy; texit_meas(i,1)];
    end

    if strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3}, 'DFW' ) && strcmp(
data{i,2}, 'TDP' )
        texit_pred_DFW_TDP_Heavy = [texit_pred_DFW_TDP_Heavy; texit_pred(i,1)];
        texit_meas_DFW_TDP_Heavy = [texit_meas_DFW_TDP_Heavy; texit_meas(i,1)];
    end

end



%%% MEM
% Get exit times - small
for i = 1:imax

    if strcmp( data{i,4}, 'Small' ) && strcmp( data{i,3}, 'MEM' ) && strcmp(
data{i,2}, 'APA' )
        texit_pred_MEM_APA_Small = [texit_pred_MEM_APA_Small; texit_pred(i,1)];
        texit_meas_MEM_APA_Small = [texit_meas_MEM_APA_Small; texit_meas(i,1)];
    end

    if strcmp( data{i,4}, 'Small' ) && strcmp( data{i,3}, 'MEM' ) && strcmp(
data{i,2}, 'TDP' )
        texit_pred_MEM_TDP_Small = [texit_pred_MEM_TDP_Small; texit_pred(i,1)];
        texit_meas_MEM_TDP_Small = [texit_meas_MEM_TDP_Small; texit_meas(i,1)];
    end

end

% Get exit times - large
for i = 1:imax

    if strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3}, 'MEM' ) && strcmp(
data{i,2}, 'APA' )
        texit_pred_MEM_APA_Large = [texit_pred_MEM_APA_Large; texit_pred(i,1)];
        texit_meas_MEM_APA_Large = [texit_meas_MEM_APA_Large; texit_meas(i,1)];
    end
```

```
    if strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3}, 'MEM' ) && strcmp(
data{i,2}, 'TDP' )
        texit_pred_MEM_TDP_Large = [texit_pred_MEM_TDP_Large; texit_pred(i,1)];
        texit_meas_MEM_TDP_Large = [texit_meas_MEM_TDP_Large; texit_meas(i,1)];
    end

end

% Get exit times - heavy
for i = 1:imax

    if strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3}, 'MEM' ) && strcmp(
data{i,2}, 'APA' )
        texit_pred_MEM_APA_Heavy = [texit_pred_MEM_APA_Heavy; texit_pred(i,1)];
        texit_meas_MEM_APA_Heavy = [texit_meas_MEM_APA_Heavy; texit_meas(i,1)];
    end

    if strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3}, 'MEM' ) && strcmp(
data{i,2}, 'TDP' )
        texit_pred_MEM_TDP_Heavy = [texit_pred_MEM_TDP_Heavy; texit_pred(i,1)];
        texit_meas_MEM_TDP_Heavy = [texit_meas_MEM_TDP_Heavy; texit_meas(i,1)];
    end

end


%%%---------------------------- PLOTS ----------------------------------%%%
% Plot exit times - DFW
disp('DFW')
figure( 'Name', 'DFW' );

temp1 = polyfit( texit_pred_DFW_APA_Small( ~isnan(texit_pred_DFW_APA_Small) ),
texit_meas_DFW_APA_Small( ~isnan(texit_pred_DFW_APA_Small) ), 1 );
temp2 = polyfit( texit_pred_DFW_TDP_Small( ~isnan(texit_pred_DFW_TDP_Small) ),
texit_meas_DFW_TDP_Small( ~isnan(texit_pred_DFW_TDP_Small) ), 1 );
subplot( 3, 1, 1 )
plot( texit_pred_DFW_APA_Small, texit_meas_DFW_APA_Small, 'ro', ...
      texit_pred_DFW_TDP_Small, texit_meas_DFW_TDP_Small, 'bo', ...
      1:180, temp1(1)*(1:180) + temp1(2), 'r--', ...
      1:180, temp2(1)*(1:180) + temp2(2), 'b--', ...
      1:180, 1:180, 'k-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
xlabel( 'Pred. Corridor Exit Time, t_{exit} (s)' )
ylabel( 'Meas. Corridor Exit Time, t_{exit} (s)' )
title( 'Small Aircraft' )
axis([0 180 0 180])
grid on

disp(['APA Small Slope = ', num2str(temp1(1))])
disp(['APA Small Inter = ', num2str(temp1(2))])
disp(['TDP Small Slope = ', num2str(temp2(1))])
disp(['TDP Small Inter = ', num2str(temp2(2))])

temp1 = polyfit( texit_pred_DFW_APA_Large( ~isnan(texit_pred_DFW_APA_Large) ),
texit_meas_DFW_APA_Large( ~isnan(texit_pred_DFW_APA_Large) ), 1 );
temp2 = polyfit( texit_pred_DFW_TDP_Large( ~isnan(texit_pred_DFW_TDP_Large) ),
texit_meas_DFW_TDP_Large( ~isnan(texit_pred_DFW_TDP_Large) ), 1 );
subplot( 3, 1, 2 )
plot( texit_pred_DFW_APA_Large, texit_meas_DFW_APA_Large, 'ro', ...
      texit_pred_DFW_TDP_Large, texit_meas_DFW_TDP_Large, 'bo', ...
      1:180, temp1(1)*(1:180) + temp1(2), 'r--', ...
      1:180, temp2(1)*(1:180) + temp2(2), 'b--', ...
      1:180, 1:180, 'k-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
xlabel( 'Pred. Corridor Exit Time, t_{exit} (s)' )
ylabel( 'Meas. Corridor Exit Time, t_{exit} (s)' )
title( 'Large Aircraft' )
```

```
axis([0 180 0 180])
grid on

disp(['APA Large Slope = ', num2str(temp1(1))])
disp(['APA Large Inter = ', num2str(temp1(2))])
disp(['TDP Large Slope = ', num2str(temp2(1))])
disp(['TDP Large Inter = ', num2str(temp2(2))])

texit_pred_DFW_APA_Heavy = texit_pred_DFW_APA_Heavy( ~isnan(texit_meas_DFW_APA_Heavy)
);
texit_meas_DFW_APA_Heavy = texit_meas_DFW_APA_Heavy( ~isnan(texit_meas_DFW_APA_Heavy)
);
temp1 = polyfit( texit_pred_DFW_APA_Heavy( ~isnan(texit_pred_DFW_APA_Heavy) ),
texit_meas_DFW_APA_Heavy( ~isnan(texit_pred_DFW_APA_Heavy) ), 1 );
texit_pred_DFW_TDP_Heavy = texit_pred_DFW_TDP_Heavy( ~isnan(texit_meas_DFW_TDP_Heavy)
);
texit_meas_DFW_TDP_Heavy = texit_meas_DFW_TDP_Heavy( ~isnan(texit_meas_DFW_TDP_Heavy)
);
temp2 = polyfit( texit_pred_DFW_TDP_Heavy( ~isnan(texit_pred_DFW_TDP_Heavy) ),
texit_meas_DFW_TDP_Heavy( ~isnan(texit_pred_DFW_TDP_Heavy) ), 1 );
subplot( 3, 1, 3 )
plot( texit_pred_DFW_APA_Heavy, texit_meas_DFW_APA_Heavy, 'ro', ...
      texit_pred_DFW_TDP_Heavy, texit_meas_DFW_TDP_Heavy, 'bo', ...
      1:180, temp1(1)*(1:180) + temp1(2), 'r--', ...
      1:180, temp2(1)*(1:180) + temp2(2), 'b--', ...
      1:180, 1:180, 'k-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
xlabel( 'Pred. Corridor Exit Time, t_{exit} (s)' )
ylabel( 'Meas. Corridor Exit Time, t_{exit} (s)' )
title( 'Heavy Aircraft' )
axis([0 180 0 180])
grid on

disp(['APA Heavy Slope = ', num2str(temp1(1))])
disp(['APA Heavy Inter = ', num2str(temp1(2))])
disp(['TDP Heavy Slope = ', num2str(temp2(1))])
disp(['TDP Heavy Inter = ', num2str(temp2(2))])

% Plot exit times - MEM
disp('MEM')
figure( 'Name', 'MEM' );

temp1 = polyfit( texit_pred_MEM_APA_Small( ~isnan(texit_pred_MEM_APA_Small) ),
texit_meas_MEM_APA_Small( ~isnan(texit_pred_MEM_APA_Small) ), 1 );
temp2 = polyfit( texit_pred_MEM_TDP_Small( ~isnan(texit_pred_MEM_TDP_Small) ),
texit_meas_MEM_TDP_Small( ~isnan(texit_pred_MEM_TDP_Small) ), 1 );
subplot( 3, 1, 1 )
plot( texit_pred_MEM_APA_Small, texit_meas_MEM_APA_Small, 'ro', ...
      texit_pred_MEM_TDP_Small, texit_meas_MEM_TDP_Small, 'bo', ...
      1:180, temp1(1)*(1:180) + temp1(2), 'r--', ...
      1:180, temp2(1)*(1:180) + temp2(2), 'b--', ...
      1:180, 1:180, 'k-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
xlabel( 'Pred. Corridor Exit Time, t_{exit} (s)' )
ylabel( 'Meas. Corridor Exit Time, t_{exit} (s)' )
title( 'Small Aircraft' )
axis([0 180 0 180])
grid on

disp(['APA Small Slope = ', num2str(temp1(1))])
disp(['APA Small Inter = ', num2str(temp1(2))])
disp(['TDP Small Slope = ', num2str(temp2(1))])
disp(['TDP Small Inter = ', num2str(temp2(2))])

temp1 = polyfit( texit_pred_MEM_APA_Large( ~isnan(texit_pred_MEM_APA_Large) ),
texit_meas_MEM_APA_Large( ~isnan(texit_pred_MEM_APA_Large) ), 1 );
temp2 = polyfit( texit_pred_MEM_TDP_Large( ~isnan(texit_pred_MEM_TDP_Large) ),
texit_meas_MEM_TDP_Large( ~isnan(texit_pred_MEM_TDP_Large) ), 1 );
```

```
subplot( 3, 1, 2 )
plot( texit_pred_MEM_APA_Large, texit_meas_MEM_APA_Large, 'ro', ...
      texit_pred_MEM_TDP_Large, texit_meas_MEM_TDP_Large, 'bo', ...
      1:180, temp1(1)*(1:180) + temp1(2), 'r--', ...
      1:180, temp2(1)*(1:180) + temp2(2), 'b--', ...
      1:180, 1:180, 'k-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
xlabel( 'Pred. Corridor Exit Time, t_{exit} (s)' )
ylabel( 'Meas. Corridor Exit Time, t_{exit} (s)' )
title( 'Large Aircraft' )
axis([0 180 0 180])
grid on

disp(['APA Large Slope = ', num2str(temp1(1))])
disp(['APA Large Inter = ', num2str(temp1(2))])
disp(['TDP Large Slope = ', num2str(temp2(1))])
disp(['TDP Large Inter = ', num2str(temp2(2))])

temp1 = polyfit( texit_pred_MEM_APA_Heavy( ~isnan(texit_pred_MEM_APA_Heavy) ),
texit_meas_MEM_APA_Heavy( ~isnan(texit_pred_MEM_APA_Heavy) ), 1 );
temp2 = polyfit( texit_pred_MEM_TDP_Heavy( ~isnan(texit_pred_MEM_TDP_Heavy) ),
texit_meas_MEM_TDP_Heavy( ~isnan(texit_pred_MEM_TDP_Heavy) ), 1 );
subplot( 3, 1, 3 )
plot( texit_pred_MEM_APA_Heavy, texit_meas_MEM_APA_Heavy, 'ro', ...
      texit_pred_MEM_TDP_Heavy, texit_meas_MEM_TDP_Heavy, 'bo', ...
      1:180, temp1(1)*(1:180) + temp1(2), 'r--', ...
      1:180, temp2(1)*(1:180) + temp2(2), 'b--', ...
      1:180, 1:180, 'k-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
xlabel( 'Pred. Corridor Exit Time, t_{exit} (s)' )
ylabel( 'Meas. Corridor Exit Time, t_{exit} (s)' )
title( 'Heavy Aircraft' )
axis([0 180 0 180])
grid on

disp(['APA Heavy Slope = ', num2str(temp1(1))])
disp(['APA Heavy Inter = ', num2str(temp1(2))])
disp(['TDP Heavy Slope = ', num2str(temp2(1))])
disp(['TDP Heavy Inter = ', num2str(temp2(2))])
```

plot_correlation_coeff.m

```
% Plot Correlation Coefficient Distributions
% This script creates box plots of the correlation coefficient
% distributions (i.e. the cross-correlation with zero lag)
% All plots are split up by location, aircraft type, and wake model.
clear all; close all; clc;


%%%------------------------ BEGIN USER INPUTS -------------------------%%%

% Load data file
load ALL_resid

%%%------------------------ END OF USER INPUTS ------------------------%%%

imax = length(data);

texit_meas = zeros( imax, 1 );
texit_pred = zeros( imax, 1 );

time   = cell(imax, 1);
acorr  = cell(imax, 3);
acorrp = cell(imax, 3);

for i = 1:imax

    clear time_meas circ_meas ypos_meas zpos_meas
    clear time_pred circ_pred ypos_pred zpos_pred

    wspan = data{i,5}; % wing span
    [rows_meas, cols_meas] = size( data{i,7} );
    [rows_pred, cols_pred] = size( data{i,8} );

    if ( rows_meas > 1 ) && ( rows_pred > 1 )

        time_meas = data{i,7}(:,1); % measured time
        circ_meas = data{i,7}(:,2); % measured circulation
        ypos_meas = data{i,7}(:,3); % measured y-pos
        zpos_meas = data{i,7}(:,4); % measured z-pos

        time_pred = data{i,8}(:,1); % predicted time 1
        circ_pred = data{i,8}(:,2); % predicted circulation 1
        ypos_pred = data{i,8}(:,3); % predicted y-pos 1
        zpos_pred = data{i,8}(:,4); % predicted z-pos 1

        time_pred100 = data{i,16}(:,1); % predicted time 100
        circ_pred100 = data{i,16}(:,2); % predicted circulation 100
        ypos_pred100 = data{i,16}(:,3); % predicted y-pos 100
        zpos_pred100 = data{i,16}(:,4); % predicted z-pos 100

        % match array sizes between measured and predicted
        ti = max( time_meas(1), time_pred(1) );
        tf = min( time_meas(end), time_pred(end) );

        im = find( time_meas == ti );
        fm = find( time_meas == tf );
        ip = find( time_pred == ti );
        fp = find( time_pred == tf );

        time_meas_adj1 = time_meas(im:fm,1);
        circ_meas_adj1 = abs( circ_meas(im:fm,1) );
        ypos_meas_adj1 = ypos_meas(im:fm,1);
        zpos_meas_adj1 = zpos_meas(im:fm,1);
```

```
        time_pred_adj1 = time_pred(ip:fp,1);
        circ_pred_adj1 = abs( circ_pred(ip:fp,1) );
        ypos_pred_adj1 = ypos_pred(ip:fp,1);
        zpos_pred_adj1 = zpos_pred(ip:fp,1);

        % match array sizes between predicted and predicted 100
        ti = max( time_pred(1), time_pred100(1) );
        tf = min( time_pred(end), time_pred100(end) );

        ip = find( time_pred == ti );
        fp = find( time_pred == tf );
        ip100 = find( time_pred100 == ti );
        fp100 = find( time_pred100 == tf );

        time_pred_adj2 = time_pred(ip:fp,1);
        circ_pred_adj2 = abs( circ_pred(ip:fp,1) );
        ypos_pred_adj2 = ypos_pred(ip:fp,1);
        zpos_pred_adj2 = zpos_pred(ip:fp,1);

        time_pred100_adj2 = time_pred100(ip100:fp100,1);
        circ_pred100_adj2 = abs( circ_pred100(ip100:fp100,1) );
        ypos_pred100_adj2 = ypos_pred100(ip100:fp100,1);
        zpos_pred100_adj2 = zpos_pred100(ip100:fp100,1);

        % get correlation for measured vs predicted data
        acorr{i,1} = abs( corr( circ_meas_adj1, circ_pred_adj1 ) );
        acorr{i,2} = abs( corr( ypos_meas_adj1, ypos_pred_adj1 ) );
        acorr{i,3} = abs( corr( zpos_meas_adj1, zpos_pred_adj1 ) );

        if isempty(acorr{i,1}); acorr{i,1} = NaN; end;
        if isempty(acorr{i,2}); acorr{i,2} = NaN; end;
        if isempty(acorr{i,3}); acorr{i,3} = NaN; end;

        % get correlation for predicted vs predicted 100 data
        acorrp{i,1} = abs( corr( circ_pred_adj2, circ_pred100_adj2 ) );
        acorrp{i,2} = abs( corr( ypos_pred_adj2, ypos_pred100_adj2 ) );
        acorrp{i,3} = abs( corr( zpos_pred_adj2, zpos_pred100_adj2 ) );

        if isempty(acorrp{i,1}); acorrp{i,1} = NaN; end;
        if isempty(acorrp{i,2}); acorrp{i,2} = NaN; end;
        if isempty(acorrp{i,3}); acorrp{i,3} = NaN; end;

    else

        acorr{i,1} = NaN;
        acorr{i,2} = NaN;
        acorr{i,3} = NaN;
        acorrp{i,1} = NaN;
        acorrp{i,2} = NaN;
        acorrp{i,3} = NaN;

    end

end

% Plot Data --------------------------------------------------------------
n = 1;

% Box Plots of Correlation (lag = 0) --------------------------------------
% Analysis Type ----------------------------------------------------------
clear G
for i = 1:imax
    G{i,1} = nominal( data{i,2} );
end

acorrmat = cell2mat(acorr);
% remove outliers
outlierflg(:,1) = ( abs( (acorrmat(:,1) - mean(~isnan(acorrmat(:,1)))) /
std(~isnan(acorrmat(:,1))) ) >  1.5 );
```

```
outlierflg(:,2) = ( abs( (acorrmat(:,2) - mean(~isnan(acorrmat(:,2)))) /
std(~isnan(acorrmat(:,2))) ) >  1.5 );
outlierflg(:,3) = ( abs( (acorrmat(:,3) - mean(~isnan(acorrmat(:,3)))) /
std(~isnan(acorrmat(:,3))) ) >  1.5 );

figure( 'Name', 'Analysis Type - P1vM' );
subplot( 2, 2, 1 )
h = boxplot( acorrmat(outlierflg(:,1),1), G(outlierflg(:,1)), 'symbol', '' );
set( h(1:6,:), 'LineWidth', 2 )
ylabel( 'Circulation Correlation' )
grid on
subplot( 2, 2, 3 )
h = boxplot( acorrmat(outlierflg(:,1),2), G(outlierflg(:,1)), 'symbol', '' );
set( h(1:6,:), 'LineWidth', 2 )
ylabel( 'Y-Position Correlation' )
grid on
subplot( 2, 2, 4 )
h = boxplot( acorrmat(outlierflg(:,1),3), G(outlierflg(:,1)), 'symbol', '' );
set( h(1:6,:), 'LineWidth', 2 )
ylabel( 'Z-Position Correlation' )
grid on

acorrmat = cell2mat(acorrp);

figure( 'Name', 'Analysis Type - P1vP100' );
subplot( 2, 2, 1 )
h = boxplot( acorrmat(outlierflg(:,1),1), G(outlierflg(:,1)), 'symbol', '' );
set( h(1:6,:), 'LineWidth', 2 )
ylabel( 'Circulation Correlation' )
V = axis; axis( [V(1) V(2) 0.95 1.05] );
grid on
subplot( 2, 2, 3 )
h = boxplot( acorrmat(outlierflg(:,1),2), G(outlierflg(:,1)), 'symbol', '' );
set( h(1:6,:), 'LineWidth', 2 )
ylabel( 'Y-Position Correlation' )
V = axis; axis( [V(1) V(2) 0.85 1.05] );
grid on
subplot( 2, 2, 4 )
h = boxplot( acorrmat(outlierflg(:,1),3), G(outlierflg(:,1)), 'symbol', '' );
set( h(1:6,:), 'LineWidth', 2 )
ylabel( 'Z-Position Correlation' )
V = axis; axis( [V(1) V(2) 0.85 1.05] );
grid on

% pause

% Location --------------------------------------------------------
clear G
for i = 1:imax
    G{i,1} = nominal( data{i,3} );
end

acorrmat = cell2mat(acorr);

figure( 'Name', 'Location - P1vM' );
subplot( 2, 2, 1 )
h = boxplot( acorrmat(:,1), G, 'symbol', '' );
set( h(1:6,:), 'LineWidth', 2 )
ylabel( 'Circulation Correlation' )
grid on
subplot( 2, 2, 3 )
h = boxplot( acorrmat(:,2), G, 'symbol', '' );
set( h(1:6,:), 'LineWidth', 2 )
ylabel( 'Y-Position Correlation' )
grid on
subplot( 2, 2, 4 )
h = boxplot( acorrmat(:,3), G, 'symbol', '' );
set( h(1:6,:), 'LineWidth', 2 )
ylabel( 'Z-Position Correlation' )
grid on
```

```
acorrmat = cell2mat(acorrp);

figure( 'Name', 'Location - P1vP100' );
subplot( 2, 2, 1 )
h = boxplot( acorrmat(:,1), G, 'symbol', '' );
set( h(1:6,:), 'LineWidth', 2 )
ylabel( 'Circulation Correlation' )
V = axis; axis( [V(1) V(2) 0.95 1.05] );
grid on
subplot( 2, 2, 3 )
h = boxplot( acorrmat(:,2), G, 'symbol', '' );
set( h(1:6,:), 'LineWidth', 2 )
ylabel( 'Y-Position Correlation' )
V = axis; axis( [V(1) V(2) 0.85 1.05] );
grid on
subplot( 2, 2, 4 )
h = boxplot( acorrmat(:,3), G, 'symbol', '' );
set( h(1:6,:), 'LineWidth', 2 )
ylabel( 'Z-Position Correlation' )
V = axis; axis( [V(1) V(2) 0.85 1.05] );
grid on

% AC Class -----------------------------------------------------------
clear G
for i = 1:imax
    G{i,1} = nominal( data{i,4} );
end

acorrmat = cell2mat(acorr);

figure( 'Name', 'AC Class - P1vM' );
subplot( 2, 2, 1 )
h = boxplot( acorrmat(:,1), G, 'symbol', '', 'grouporder', {'Heavy','Large','Small'}
);
set( h(1:6,:), 'LineWidth', 2 )
ylabel( 'Circulation Correlation' )
grid on
subplot( 2, 2, 3 )
h = boxplot( acorrmat(:,2), G, 'symbol', '', 'grouporder', {'Heavy','Large','Small'}
);
set( h(1:6,:), 'LineWidth', 2 )
ylabel( 'Y-Position Correlation' )
grid on
subplot( 2, 2, 4 )
h = boxplot( acorrmat(:,3), G, 'symbol', '', 'grouporder', {'Heavy','Large','Small'}
);
set( h(1:6,:), 'LineWidth', 2 )
ylabel( 'Z-Position Correlation' )
grid on

acorrmat = cell2mat(acorrp);

figure( 'Name', 'AC Class - P1vP100' );
subplot( 2, 2, 1 )
h = boxplot( acorrmat(:,1), G, 'symbol', '', 'grouporder', {'Heavy','Large','Small'}
);
set( h(1:6,:), 'LineWidth', 2 )
ylabel( 'Circulation Correlation' )
V = axis; axis( [V(1) V(2) 0.95 1.05] );
grid on
subplot( 2, 2, 3 )
h = boxplot( acorrmat(:,2), G, 'symbol', '', 'grouporder', {'Heavy','Large','Small'}
);
set( h(1:6,:), 'LineWidth', 2 )
ylabel( 'Y-Position Correlation' )
V = axis; axis( [V(1) V(2) 0.85 1.05] );
grid on
subplot( 2, 2, 4 )
```

```
h = boxplot( acorrmat(:,3), G, 'symbol', '', 'grouporder', {'Heavy','Large','Small'}
);
set( h(1:6,:), 'LineWidth', 2 )
ylabel( 'Z-Position Correlation' )
V = axis; axis( [V(1) V(2) 0.85 1.05] );
grid on

% Analysis Type and AC Class ---------------------------------------------
clear G
% for i = 1:imax
%     G1{i,1} = nominal( data{i,2} );
%     G2{i,1} = nominal( data{i,4} );
% end
G{1} = data(:,2);
G{2} = data(:,4);

acorrmat = cell2mat(acorr);

figure( 'Name', 'AC Class - P1vM' );
subplot( 2, 2, 1 )
h = boxplot( acorrmat(:,1), G, 'symbol', '', 'grouporder',
{'APA,Heavy','TDP,Heavy','APA,Large','TDP,Large','APA,Small','TDP,Small'} );
set( h(1:6,:), 'LineWidth', 2 )
ylabel( 'Circulation Correlation' )
grid on
subplot( 2, 2, 3 )
h = boxplot( acorrmat(:,2), G, 'symbol', '', 'grouporder',
{'APA,Heavy','TDP,Heavy','APA,Large','TDP,Large','APA,Small','TDP,Small'} );
set( h(1:6,:), 'LineWidth', 2 )
ylabel( 'Y-Position Correlation' )
grid on
subplot( 2, 2, 4 )
h = boxplot( acorrmat(:,3), G, 'symbol', '', 'grouporder',
{'APA,Heavy','TDP,Heavy','APA,Large','TDP,Large','APA,Small','TDP,Small'} );
set( h(1:6,:), 'LineWidth', 2 )
ylabel( 'Z-Position Correlation' )
grid on

figure( 'Name', 'AC Class - P1vM 1' );
h = boxplot( acorrmat(:,1), G, 'symbol', '', 'grouporder',
{'APA,Heavy','TDP,Heavy','APA,Large','TDP,Large','APA,Small','TDP,Small'} );
set( h(1:6,:), 'LineWidth', 2 )
ylabel( 'Circulation Correlation' )
grid on
figure( 'Name', 'AC Class - P1vM 2' );
h = boxplot( acorrmat(:,2), G, 'symbol', '', 'grouporder',
{'APA,Heavy','TDP,Heavy','APA,Large','TDP,Large','APA,Small','TDP,Small'} );
set( h(1:6,:), 'LineWidth', 2 )
ylabel( 'Y-Position Correlation' )
grid on
figure( 'Name', 'AC Class - P1vM 3' );
h = boxplot( acorrmat(:,3), G, 'symbol', '', 'grouporder',
{'APA,Heavy','TDP,Heavy','APA,Large','TDP,Large','APA,Small','TDP,Small'} );
set( h(1:6,:), 'LineWidth', 2 )
ylabel( 'Z-Position Correlation' )
grid on

acorrmat = cell2mat(acorrp);

figure( 'Name', 'AC Class - P1vP100' );
subplot( 2, 2, 1 )
h = boxplot( acorrmat(:,1), G, 'symbol', '', 'grouporder',
{'APA,Heavy','TDP,Heavy','APA,Large','TDP,Large','APA,Small','TDP,Small'} );
set( h(1:6,:), 'LineWidth', 2 )
ylabel( 'Circulation Correlation' )
V = axis; axis( [V(1) V(2) 0.95 1.05] );
grid on
subplot( 2, 2, 3 )
h = boxplot( acorrmat(:,2), G, 'symbol', '', 'grouporder',
{'APA,Heavy','TDP,Heavy','APA,Large','TDP,Large','APA,Small','TDP,Small'} );
```

```
set( h(1:6,:), 'LineWidth', 2 )
ylabel( 'Y-Position Correlation' )
V = axis; axis( [V(1) V(2) 0.85 1.05] );
grid on
subplot( 2, 2, 4 )
h = boxplot( acorrmat(:,3), G, 'symbol', '', 'grouporder',
{'APA,Heavy','TDP,Heavy','APA,Large','TDP,Large','APA,Small','TDP,Small'} );
set( h(1:6,:), 'LineWidth', 2 )
ylabel( 'Z-Position Correlation' )
V = axis; axis( [V(1) V(2) 0.85 1.05] );
grid on

% END BOX PLOTS OF CORRELATION --------------------------------------------
```

plot_cross_correlation.m

```
% Plot Cross-Correlation vs. Lag Time
% This script plots the mean cross-correlation vs. time. If you would like
% to plot the actual cross-correlation vs. time, uncomment the last section
% of plots at the bottom of this script. NOTE: This will plot ALL the data
% and will be memory intensive.
% All plots are split up by location, aircraft type, and wake model.
clear all; close all; clc;


%%%------------------------ BEGIN USER INPUTS --------------------------%%%

% Load data file
load ALL_resid

%%%------------------------ END OF USER INPUTS -------------------------%%%


xstring = 'Lag Time, l (s)';
speedms = 1;

% imax = 1509;
imax = length(data);

texit_meas = zeros( imax, 1 );
texit_pred = zeros( imax, 1 );

time   = cell(imax, 1);
acorr  = cell(imax, 3);
acorrp = cell(imax, 3);

for i = 1:imax

    clear time_meas circ_meas ypos_meas zpos_meas
    clear time_pred circ_pred ypos_pred zpos_pred

    wspan = data{i,5}; % wing span
    [rows_meas, cols_meas] = size( data{i,7} );
    [rows_pred, cols_pred] = size( data{i,8} );

    if ( rows_meas > 1 ) && ( rows_pred > 1 )

        time_meas = data{i,7}(:,1); % measured time
        circ_meas = data{i,7}(:,2); % measured circulation
        ypos_meas = data{i,7}(:,3); % measured y-pos
        zpos_meas = data{i,7}(:,4); % measured z-pos

        time_pred = data{i,8}(:,1); % predicted time 1m res.
        circ_pred = data{i,8}(:,2); % predicted circulation 1m res.
        ypos_pred = data{i,8}(:,3); % predicted y-pos 1m res.
        zpos_pred = data{i,8}(:,4); % predicted z-pos 1m res.

        time_pred100 = data{i,16}(:,1); % predicted time 100m res.
        circ_pred100 = data{i,16}(:,2); % predicted circulation 100m res.
        ypos_pred100 = data{i,16}(:,3); % predicted y-pos 100m res.
        zpos_pred100 = data{i,16}(:,4); % predicted z-pos 100m res.

        % match array sizes between measured and predicted
        ti = max( time_meas(1), time_pred(1) );
        tf = min( time_meas(end), time_pred(end) );

        im = find( time_meas == ti );
        fm = find( time_meas == tf );
        ip = find( time_pred == ti );
        fp = find( time_pred == tf );

         % get adjusted measured and predicted data sets
        time_meas_adj1 = time_meas(im:fm,1);
        circ_meas_adj1 = abs( circ_meas(im:fm,1) );
        ypos_meas_adj1 = ypos_meas(im:fm,1);
```

```
        zpos_meas_adj1 = zpos_meas(im:fm,1);

        time_pred_adj1 = time_pred(ip:fp,1);
        circ_pred_adj1 = abs( circ_pred(ip:fp,1) );
        ypos_pred_adj1 = ypos_pred(ip:fp,1);
        zpos_pred_adj1 = zpos_pred(ip:fp,1);

        % match array sizes between predicted and predicted 100
        ti = max( time_pred(1), time_pred100(1) );
        tf = min( time_pred(end), time_pred100(end) );

        ip = find( time_pred == ti );
        fp = find( time_pred == tf );
        ip100 = find( time_pred100 == ti );
        fp100 = find( time_pred100 == tf );

         % get adjusted predicted data sets
        time_pred_adj2 = time_pred(ip:fp,1);
        circ_pred_adj2 = abs( circ_pred(ip:fp,1) );
        ypos_pred_adj2 = ypos_pred(ip:fp,1);
        zpos_pred_adj2 = zpos_pred(ip:fp,1);

        time_pred100_adj2 = time_pred100(ip100:fp100,1);
        circ_pred100_adj2 = abs( circ_pred100(ip100:fp100,1) );
        ypos_pred100_adj2 = ypos_pred100(ip100:fp100,1);
        zpos_pred100_adj2 = zpos_pred100(ip100:fp100,1);

        % get cross correlation for measured vs predicted data
        time{i,1}  = time_meas_adj1;
        [c_ww,lags] = xcorr( circ_meas_adj1, circ_pred_adj1, 'coeff' ); % circulation
        acorr{i,1} = [c_ww,lags'];
        [c_ww,lags] = xcorr( ypos_meas_adj1, ypos_pred_adj1, 'coeff' ); % y-pos
        acorr{i,2} = [c_ww,lags'];
        [c_ww,lags] = xcorr( zpos_meas_adj1, zpos_pred_adj1, 'coeff' ); % z-pos
        acorr{i,3} = [c_ww,lags'];

        if acorr{i,1}( ceil(end/2), 1 ) < 0.0
            acorr{i,1}(:,1) = -acorr{i,1}(:,1);
        end
        if acorr{i,2}( ceil(end/2), 1 ) < 0.0
            acorr{i,2}(:,1) = -acorr{i,2}(:,1);
        end
        if acorr{i,3}( ceil(end/2), 1 ) < 0.0
            acorr{i,3}(:,1) = -acorr{i,3}(:,1);
        end

        % remove correlations that go over 1 (outliers)
        if sum( acorr{i,1}(:,1) > 1 | acorr{i,1}(:,1) < 0 ) ~= 0
           acorr{i,1}(:,1) = NaN;
        end
        if sum( acorr{i,2}(:,1) > 1 | acorr{i,2}(:,1) < 0 ) ~= 0
           acorr{i,2}(:,1) = NaN;
        end
        if sum( acorr{i,3}(:,1) > 1 | acorr{i,3}(:,1) < 0 ) ~= 0
           acorr{i,3}(:,1) = NaN;
        end

        % get cross correlation for predicted vs predicted 100 data
        time{i,1}  = time_pred_adj2;
        [c_ww,lags] = xcorr( circ_pred_adj2, circ_pred100_adj2, 'coeff' ); %
circulation
        acorrp{i,1} = [c_ww,lags'];
        [c_ww,lags] = xcorr( ypos_pred_adj2, ypos_pred100_adj2, 'coeff' ); % y-pos
        acorrp{i,2} = [c_ww,lags'];
        [c_ww,lags] = xcorr( zpos_pred_adj2, zpos_pred100_adj2, 'coeff' ); % z-pos
        acorrp{i,3} = [c_ww,lags'];

        % remove correlations that go over 1 (outliers)
        if sum( acorrp{i,1}(:,1) > 1 | acorrp{i,1}(:,1) < 0 | acorrp{i,1}(:,2) > 180 )
~= 0
```

```
            acorrp{i,1}(:,1) = NaN;
        end
        if sum( acorrp{i,2}(:,1) > 1 | acorrp{i,2}(:,1) < 0 | acorrp{i,2}(:,2) > 180 )
~= 0
            acorrp{i,2}(:,1) = NaN;
        end
        if sum( acorrp{i,3}(:,1) > 1 | acorrp{i,3}(:,1) < 0 | acorrp{i,3}(:,2) > 180 )
~= 0
            acorrp{i,3}(:,1) = NaN;
        end

    end

end

% Plot Cross-Correlation vs. Lag Time (Measured vs. Predicted 1m) ---------
n = 1;

small_dfw_c_apa(:,1) = (-200:200)';
small_dfw_c_tdp(:,1) = (-200:200)';
small_dfw_y_apa(:,1) = (-200:200)';
small_dfw_y_tdp(:,1) = (-200:200)';
small_dfw_z_apa(:,1) = (-200:200)';
small_dfw_z_tdp(:,1) = (-200:200)';
small_mem_c_apa(:,1) = (-200:200)';
small_mem_c_tdp(:,1) = (-200:200)';
small_mem_y_apa(:,1) = (-200:200)';
small_mem_y_tdp(:,1) = (-200:200)';
small_mem_z_apa(:,1) = (-200:200)';
small_mem_z_tdp(:,1) = (-200:200)';

large_dfw_c_apa(:,1) = (-200:200)';
large_dfw_c_tdp(:,1) = (-200:200)';
large_dfw_y_apa(:,1) = (-200:200)';
large_dfw_y_tdp(:,1) = (-200:200)';
large_dfw_z_apa(:,1) = (-200:200)';
large_dfw_z_tdp(:,1) = (-200:200)';
large_mem_c_apa(:,1) = (-200:200)';
large_mem_c_tdp(:,1) = (-200:200)';
large_mem_y_apa(:,1) = (-200:200)';
large_mem_y_tdp(:,1) = (-200:200)';
large_mem_z_apa(:,1) = (-200:200)';
large_mem_z_tdp(:,1) = (-200:200)';

heavy_dfw_c_apa(:,1) = (-200:200)';
heavy_dfw_c_tdp(:,1) = (-200:200)';
heavy_dfw_y_apa(:,1) = (-200:200)';
heavy_dfw_y_tdp(:,1) = (-200:200)';
heavy_dfw_z_apa(:,1) = (-200:200)';
heavy_dfw_z_tdp(:,1) = (-200:200)';
heavy_mem_c_apa(:,1) = (-200:200)';
heavy_mem_c_tdp(:,1) = (-200:200)';
heavy_mem_y_apa(:,1) = (-200:200)';
heavy_mem_y_tdp(:,1) = (-200:200)';
heavy_mem_z_apa(:,1) = (-200:200)';
heavy_mem_z_tdp(:,1) = (-200:200)';

%%% SMALL - DFW
% Plot Cross-correlation - small - circ
for i = 1:n:imax

    % DFW - small - circulation
    if ~isempty( acorr{i,1} ) && strcmp( data{i,4}, 'Small' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'APA' )

        foo1 = small_dfw_c_apa;
        foot = acorr{i,1}(1,2);
        foo2 = acorr{i,1}(:,1);

        [rows,cols] = size( foo2 );
```

```
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        small_dfw_c_apa = foo1;

    end

    if ~isempty( acorr{i,1} ) && strcmp( data{i,4}, 'Small' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = small_dfw_c_tdp;
        foot = acorr{i,1}(1,2);
        foo2 = acorr{i,1}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        small_dfw_c_tdp = foo1;

    end

    % DFW - small - y
    if ~isempty( acorr{i,2} ) && strcmp( data{i,4}, 'Small' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'APA' )

        foo1 = small_dfw_y_apa;
        foot = acorr{i,2}(1,2);
        foo2 = acorr{i,2}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        small_dfw_y_apa = foo1;

    end

    if ~isempty( acorr{i,2} ) && strcmp( data{i,4}, 'Small' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = small_dfw_y_tdp;
        foot = acorr{i,2}(1,2);
        foo2 = acorr{i,2}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        small_dfw_y_tdp = foo1;

    end

    % DFW - small - z
    if ~isempty( acorr{i,3} ) && strcmp( data{i,4}, 'Small' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'APA' )

        foo1 = small_dfw_z_apa;
        foot = acorr{i,3}(1,2);
        foo2 = acorr{i,3}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        small_dfw_z_apa = foo1;

    end
```

```
    if ~isempty( acorr{i,3} ) && strcmp( data{i,4}, 'Small' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = small_dfw_z_tdp;
        foot = acorr{i,3}(1,2);
        foo2 = acorr{i,3}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        small_dfw_z_tdp = foo1;

    end

    % MEM - small - circulation
    if ~isempty( acorr{i,1} ) && strcmp( data{i,4}, 'Small' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'APA' )

        foo1 = small_mem_c_apa;
        foot = acorr{i,1}(1,2);
        foo2 = acorr{i,1}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        small_mem_c_apa = foo1;

    end

    if ~isempty( acorr{i,1} ) && strcmp( data{i,4}, 'Small' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = small_mem_c_tdp;
        foot = acorr{i,1}(1,2);
        foo2 = acorr{i,1}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        small_mem_c_tdp = foo1;

    end

    % MEM acorruals - small - y
    if ~isempty( acorr{i,2} ) && strcmp( data{i,4}, 'Small' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'APA' )

        foo1 = small_mem_y_apa;
        foot = acorr{i,2}(1,2);
        foo2 = acorr{i,2}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        small_mem_y_apa = foo1;

    end

    if ~isempty( acorr{i,2} ) && strcmp( data{i,4}, 'Small' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = small_mem_y_tdp;
        foot = acorr{i,2}(1,2);
        foo2 = acorr{i,2}(:,1);

        [rows,cols] = size( foo2 );
```

```
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        small_mem_y_tdp = foo1;

    end

    % MEM - small - z
    if ~isempty( acorr{i,3} ) && strcmp( data{i,4}, 'Small' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'APA' )

        foo1 = small_mem_z_apa;
        foot = acorr{i,3}(1,2);
        foo2 = acorr{i,3}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        small_mem_z_apa = foo1;

    end

    if ~isempty( acorr{i,2} ) && strcmp( data{i,4}, 'Small' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = small_mem_z_tdp;
        foot = acorr{i,3}(1,2);
        foo2 = acorr{i,3}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        small_mem_z_tdp = foo1;

    end

    % DFW - large - circulation
    if ~isempty( acorr{i,1} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'APA' )

        foo1 = large_dfw_c_apa;
        foot = acorr{i,1}(1,2);
        foo2 = acorr{i,1}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        large_dfw_c_apa = foo1;

    end

    if ~isempty( acorr{i,1} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = large_dfw_c_tdp;
        foot = acorr{i,1}(1,2);
        foo2 = acorr{i,1}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        large_dfw_c_tdp = foo1;

    end

    % DFW - large - y
```

```
    if ~isempty( acorr{i,2} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'APA' )

        foo1 = large_dfw_y_apa;
        foot = acorr{i,2}(1,2);
        foo2 = acorr{i,2}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        large_dfw_y_apa = foo1;

    end

    if ~isempty( acorr{i,2} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = large_dfw_y_tdp;
        foot = acorr{i,2}(1,2);
        foo2 = acorr{i,2}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        large_dfw_y_tdp = foo1;

    end

    % DFW - large - z
    if ~isempty( acorr{i,3} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'APA' )

        foo1 = large_dfw_z_apa;
        foot = acorr{i,3}(1,2);
        foo2 = acorr{i,3}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        large_dfw_z_apa = foo1;

    end

    if ~isempty( acorr{i,2} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = large_dfw_z_tdp;
        foot = acorr{i,3}(1,2);
        foo2 = acorr{i,3}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        large_dfw_z_tdp = foo1;

    end

    % MEM - large - circulation
    if ~isempty( acorr{i,1} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'APA' )

        foo1 = large_mem_c_apa;
        foot = acorr{i,1}(1,2);
        foo2 = acorr{i,1}(:,1);

        [rows,cols] = size( foo2 );
```

```
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        large_mem_c_apa = foo1;

    end

    if ~isempty( acorr{i,1} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = large_mem_c_tdp;
        foot = acorr{i,1}(1,2);
        foo2 = acorr{i,1}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        large_mem_c_tdp = foo1;

    end

    % MEM - large - y
    if ~isempty( acorr{i,2} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'APA' )

        foo1 = large_mem_y_apa;
        foot = acorr{i,2}(1,2);
        foo2 = acorr{i,2}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        large_mem_y_apa = foo1;

    end

    if ~isempty( acorr{i,2} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = large_mem_y_tdp;
        foot = acorr{i,2}(1,2);
        foo2 = acorr{i,2}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        large_mem_y_tdp = foo1;

    end

    % MEM - large - z
    if ~isempty( acorr{i,3} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'APA' )

        foo1 = large_mem_z_apa;
        foot = acorr{i,3}(1,2);
        foo2 = acorr{i,3}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        large_mem_z_apa = foo1;

    end
```

```
    if ~isempty( acorr{i,3} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = large_mem_z_tdp;
        foot = acorr{i,3}(1,2);
        foo2 = acorr{i,3}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        large_mem_z_tdp = foo1;

    end

    % DFW - heavy - circulation
    if ~isempty( acorr{i,1} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'APA' )

        foo1 = heavy_dfw_c_apa;
        foot = acorr{i,1}(1,2);
        foo2 = acorr{i,1}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        heavy_dfw_c_apa = foo1;

    end

    if ~isempty( acorr{i,1} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = heavy_dfw_c_tdp;
        foot = acorr{i,1}(1,2);
        foo2 = acorr{i,1}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        heavy_dfw_c_tdp = foo1;

    end

    % DFW - heavy - y
    if ~isempty( acorr{i,2} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'APA' )

        foo1 = heavy_dfw_y_apa;
        foot = acorr{i,2}(1,2);
        foo2 = acorr{i,2}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        heavy_dfw_y_apa = foo1;

    end

    if ~isempty( acorr{i,2} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = heavy_dfw_y_tdp;
        foot = acorr{i,2}(1,2);
        foo2 = acorr{i,2}(:,1);

        [rows,cols] = size( foo2 );
```

```
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        heavy_dfw_y_tdp = foo1;

    end

    % DFW - heavy - z
    if ~isempty( acorr{i,3} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'APA' )

        foo1 = heavy_dfw_z_apa;
        foot = acorr{i,3}(1,2);
        foo2 = acorr{i,3}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        heavy_dfw_z_apa = foo1;

    end

    if ~isempty( acorr{i,3} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = heavy_dfw_z_tdp;
        foot = acorr{i,3}(1,2);
        foo2 = acorr{i,3}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        heavy_dfw_z_tdp = foo1;

    end

    % MEM - heavy - circulation
    if ~isempty( acorr{i,1} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'APA' )

        foo1 = heavy_mem_c_apa;
        foot = acorr{i,1}(1,2);
        foo2 = acorr{i,1}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        heavy_mem_c_apa = foo1;

    end

    if ~isempty( acorr{i,1} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = heavy_mem_c_tdp;
        foot = acorr{i,1}(1,2);
        foo2 = acorr{i,1}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        heavy_mem_c_tdp = foo1;

    end

    % MEM - heavy - y
```

```
    if ~isempty( acorr{i,2} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'APA' )

        foo1 = heavy_mem_y_apa;
        foot = acorr{i,2}(1,2);
        foo2 = acorr{i,2}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        heavy_mem_y_apa = foo1;

    end

    if ~isempty( acorr{i,2} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = heavy_mem_y_tdp;
        foot = acorr{i,2}(1,2);
        foo2 = acorr{i,2}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        heavy_mem_y_tdp = foo1;

    end

    % MEM - heavy - z
    if ~isempty( acorr{i,3} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'APA' )

        foo1 = heavy_mem_z_apa;
        foot = acorr{i,3}(1,2);
        foo2 = acorr{i,3}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        heavy_mem_z_apa = foo1;

    end

    if ~isempty( acorr{i,3} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = heavy_mem_z_tdp;
        foot = acorr{i,3}(1,2);
        foo2 = acorr{i,3}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        heavy_mem_z_tdp = foo1;

    end

end

% Find Min, Mean, Max across all data sets at each lag time
varlabels = { 'small_dfw_c_apa', 'small_dfw_c_tdp', 'small_dfw_y_apa',
'small_dfw_y_tdp', 'small_dfw_z_apa', 'small_dfw_z_tdp', ...
              'small_mem_c_apa', 'small_mem_c_tdp', 'small_mem_y_apa',
'small_mem_y_tdp', 'small_mem_z_apa', 'small_mem_z_tdp', ...
              'large_dfw_c_apa', 'large_dfw_c_tdp', 'large_dfw_y_apa',
'large_dfw_y_tdp', 'large_dfw_z_apa', 'large_dfw_z_tdp', ...
```

```
                'large_mem_c_apa', 'large_mem_c_tdp', 'large_mem_y_apa',
'large_mem_y_tdp', 'large_mem_z_apa', 'large_mem_z_tdp', ...
                'heavy_dfw_c_apa', 'heavy_dfw_c_tdp', 'heavy_dfw_y_apa',
'heavy_dfw_y_tdp', 'heavy_dfw_z_apa', 'heavy_dfw_z_tdp', ...
                'heavy_mem_c_apa', 'heavy_mem_c_tdp', 'heavy_mem_y_apa',
'heavy_mem_y_tdp', 'heavy_mem_z_apa', 'heavy_mem_z_tdp' };

for i = 1:length( varlabels )

    eval( ['mmm_', varlabels{i}, '(:,2) = nanmean( ', varlabels{i}, '(:,2:end), 2 );']
);
    eval( ['mmm_', varlabels{i}, '(:,4) =  nanstd( ', varlabels{i}, '(:,2:end), 0, 2
);'] );

    eval( ['mmm_', varlabels{i}, '(:,1) =  mmm_', varlabels{i}, '(:,2) - 3*mmm_',
varlabels{i}, '(:,4);'] );
    eval( ['mmm_', varlabels{i}, '(:,3) =  mmm_', varlabels{i}, '(:,2) + 3*mmm_',
varlabels{i}, '(:,4);'] );

end


% Plot Results
% SMALL ------------------------------------------------------------------
figure( 'Name', 'Small - C' )
subplot( 1, 2, 1 )
plot( (foo1(:,1)-1)*speedms, mmm_small_dfw_c_apa(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_small_dfw_c_tdp(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_small_dfw_c_apa(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_small_dfw_c_apa(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_small_dfw_c_tdp(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_small_dfw_c_tdp(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Circulation Cross-Correlation, DFW'; 'Measured vs. 1m Predicted'} )
axis( [0 200 0 2] )

subplot( 1, 2, 2 )
plot( (foo1(:,1)-1)*speedms, mmm_small_mem_c_apa(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_small_mem_c_tdp(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_small_mem_c_apa(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_small_mem_c_apa(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_small_mem_c_tdp(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_small_mem_c_tdp(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Circulation Cross-Correlation, MEM'; 'Measured vs. 1m Predicted'} )
axis( [0 200 0 2] )


figure( 'Name', 'Small - Y' )
subplot( 1, 2, 1 )
plot( (foo1(:,1)-1)*speedms, mmm_small_dfw_y_apa(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_small_dfw_y_tdp(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_small_dfw_y_apa(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_small_dfw_y_apa(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_small_dfw_y_tdp(:,1), 'bv-', ...
```

```
      (foo1(:,1)-1)*speedms, mmm_small_dfw_y_tdp(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Y-Pos. Cross-Correlation, DFW'; 'Measured vs. 1m Predicted'} )
axis( [0 200 0 2] )

subplot( 1, 2, 2 )
plot( (foo1(:,1)-1)*speedms, mmm_small_mem_y_apa(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_small_mem_y_tdp(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_small_mem_y_apa(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_small_mem_y_apa(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_small_mem_y_tdp(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_small_mem_y_tdp(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Y-Pos. Cross-Correlation, MEM'; 'Measured vs. 1m Predicted'} )
axis( [0 200 0 2] )


figure( 'Name', 'Small - Z' )
subplot( 1, 2, 1 )
plot( (foo1(:,1)-1)*speedms, mmm_small_dfw_z_apa(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_small_dfw_z_tdp(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_small_dfw_z_apa(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_small_dfw_z_apa(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_small_dfw_z_tdp(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_small_dfw_z_tdp(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Z-Pos. Cross-Correlation, DFW'; 'Measured vs. 1m Predicted'} )
axis( [0 200 0 2] )

subplot( 1, 2, 2 )
plot( (foo1(:,1)-1)*speedms, mmm_small_mem_z_apa(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_small_mem_z_tdp(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_small_mem_z_apa(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_small_mem_z_apa(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_small_mem_z_tdp(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_small_mem_z_tdp(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Z-Pos. Cross-Correlation, MEM'; 'Measured vs. 1m Predicted'} )
axis( [0 200 0 2] )


% LARGE -----------------------------------------------------------------
figure( 'Name', 'large - C' )
subplot( 1, 2, 1 )
plot( (foo1(:,1)-1)*speedms, mmm_large_dfw_c_apa(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_large_dfw_c_tdp(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
```

```
plot( (foo1(:,1)-1)*speedms, mmm_large_dfw_c_apa(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_large_dfw_c_apa(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_large_dfw_c_tdp(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_large_dfw_c_tdp(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Circulation Cross-Correlation, DFW'; 'Measured vs. 1m Predicted'} )
axis( [0 200 0 2] )

subplot( 1, 2, 2 )
plot( (foo1(:,1)-1)*speedms, mmm_large_mem_c_apa(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_large_mem_c_tdp(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_large_mem_c_apa(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_large_mem_c_apa(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_large_mem_c_tdp(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_large_mem_c_tdp(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Circulation Cross-Correlation, MEM'; 'Measured vs. 1m Predicted'} )
axis( [0 200 0 2] )


figure( 'Name', 'large - Y' )
subplot( 1, 2, 1 )
plot( (foo1(:,1)-1)*speedms, mmm_large_dfw_y_apa(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_large_dfw_y_tdp(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_large_dfw_y_apa(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_large_dfw_y_apa(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_large_dfw_y_tdp(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_large_dfw_y_tdp(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Y-Pos. Cross-Correlation, DFW'; 'Measured vs. 1m Predicted'} )
axis( [0 200 0 2] )

subplot( 1, 2, 2 )
plot( (foo1(:,1)-1)*speedms, mmm_large_mem_y_apa(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_large_mem_y_tdp(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_large_mem_y_apa(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_large_mem_y_apa(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_large_mem_y_tdp(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_large_mem_y_tdp(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Y-Pos. Cross-Correlation, MEM'; 'Measured vs. 1m Predicted'} )
axis( [0 200 0 2] )


figure( 'Name', 'large - Z' )
subplot( 1, 2, 1 )
plot( (foo1(:,1)-1)*speedms, mmm_large_dfw_z_apa(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_large_dfw_z_tdp(:,2), 'b.-', ...
      'linewidth', 2 )
```

```
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_large_dfw_z_apa(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_large_dfw_z_apa(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_large_dfw_z_tdp(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_large_dfw_z_tdp(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Z-Pos. Cross-Correlation, DFW'; 'Measured vs. 1m Predicted'} )
axis( [0 200 0 2] )

subplot( 1, 2, 2 )
plot( (foo1(:,1)-1)*speedms, mmm_large_mem_z_apa(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_large_mem_z_tdp(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_large_mem_z_apa(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_large_mem_z_apa(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_large_mem_z_tdp(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_large_mem_z_tdp(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Z-Pos. Cross-Correlation, MEM'; 'Measured vs. 1m Predicted'} )
axis( [0 200 0 2] )


% HEAVY -----------------------------------------------------------------
figure( 'Name', 'heavy - C' )
subplot( 1, 2, 1 )
plot( (foo1(:,1)-1)*speedms, mmm_heavy_dfw_c_apa(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_dfw_c_tdp(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_heavy_dfw_c_apa(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_dfw_c_apa(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_dfw_c_tdp(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_dfw_c_tdp(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Circulation Cross-Correlation, DFW'; 'Measured vs. 1m Predicted'} )
axis( [0 200 0 2] )

subplot( 1, 2, 2 )
plot( (foo1(:,1)-1)*speedms, mmm_heavy_mem_c_apa(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_mem_c_tdp(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_heavy_mem_c_apa(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_mem_c_apa(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_mem_c_tdp(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_mem_c_tdp(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Circulation Cross-Correlation, MEM'; 'Measured vs. 1m Predicted'} )
axis( [0 200 0 2] )


figure( 'Name', 'heavy - Y' )
subplot( 1, 2, 1 )
```

```
plot( (foo1(:,1)-1)*speedms, mmm_heavy_dfw_y_apa(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_dfw_y_tdp(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_heavy_dfw_y_apa(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_dfw_y_apa(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_dfw_y_tdp(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_dfw_y_tdp(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Y-Pos. Cross-Correlation, DFW'; 'Measured vs. 1m Predicted'} )
axis( [0 200 0 2] )

subplot( 1, 2, 2 )
plot( (foo1(:,1)-1)*speedms, mmm_heavy_mem_y_apa(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_mem_y_tdp(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_heavy_mem_y_apa(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_mem_y_apa(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_mem_y_tdp(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_mem_y_tdp(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Y-Pos. Cross-Correlation, MEM'; 'Measured vs. 1m Predicted'} )
axis( [0 200 0 2] )


figure( 'Name', 'heavy - Z' )
subplot( 1, 2, 1 )
plot( (foo1(:,1)-1)*speedms, mmm_heavy_dfw_z_apa(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_dfw_z_tdp(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_heavy_dfw_z_apa(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_dfw_z_apa(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_dfw_z_tdp(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_dfw_z_tdp(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Z-Pos. Cross-Correlation, DFW'; 'Measured vs. 1m Predicted'} )
axis( [0 200 0 2] )

subplot( 1, 2, 2 )
plot( (foo1(:,1)-1)*speedms, mmm_heavy_mem_z_apa(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_mem_z_tdp(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_heavy_mem_z_apa(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_mem_z_apa(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_mem_z_tdp(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_mem_z_tdp(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Z-Pos. Cross-Correlation, MEM'; 'Measured vs. 1m Predicted'} )
axis( [0 200 0 2] )
```

```
% Plot Cross-Correlation vs. Lag Time (Predicted 1m vs. Predicted 100m) ---
n = 1;

small_dfw_c_apa_p(:,1) = (-200:200)';
small_dfw_c_tdp_p(:,1) = (-200:200)';
small_dfw_y_apa_p(:,1) = (-200:200)';
small_dfw_y_tdp_p(:,1) = (-200:200)';
small_dfw_z_apa_p(:,1) = (-200:200)';
small_dfw_z_tdp_p(:,1) = (-200:200)';
small_mem_c_apa_p(:,1) = (-200:200)';
small_mem_c_tdp_p(:,1) = (-200:200)';
small_mem_y_apa_p(:,1) = (-200:200)';
small_mem_y_tdp_p(:,1) = (-200:200)';
small_mem_z_apa_p(:,1) = (-200:200)';
small_mem_z_tdp_p(:,1) = (-200:200)';

large_dfw_c_apa_p(:,1) = (-200:200)';
large_dfw_c_tdp_p(:,1) = (-200:200)';
large_dfw_y_apa_p(:,1) = (-200:200)';
large_dfw_y_tdp_p(:,1) = (-200:200)';
large_dfw_z_apa_p(:,1) = (-200:200)';
large_dfw_z_tdp_p(:,1) = (-200:200)';
large_mem_c_apa_p(:,1) = (-200:200)';
large_mem_c_tdp_p(:,1) = (-200:200)';
large_mem_y_apa_p(:,1) = (-200:200)';
large_mem_y_tdp_p(:,1) = (-200:200)';
large_mem_z_apa_p(:,1) = (-200:200)';
large_mem_z_tdp_p(:,1) = (-200:200)';

heavy_dfw_c_apa_p(:,1) = (-200:200)';
heavy_dfw_c_tdp_p(:,1) = (-200:200)';
heavy_dfw_y_apa_p(:,1) = (-200:200)';
heavy_dfw_y_tdp_p(:,1) = (-200:200)';
heavy_dfw_z_apa_p(:,1) = (-200:200)';
heavy_dfw_z_tdp_p(:,1) = (-200:200)';
heavy_mem_c_apa_p(:,1) = (-200:200)';
heavy_mem_c_tdp_p(:,1) = (-200:200)';
heavy_mem_y_apa_p(:,1) = (-200:200)';
heavy_mem_y_tdp_p(:,1) = (-200:200)';
heavy_mem_z_apa_p(:,1) = (-200:200)';
heavy_mem_z_tdp_p(:,1) = (-200:200)';

%%% SMALL - DFW
% Plot Cross-correlation - small - circ
for i = 1:n:imax

    % DFW - small - circulation
    if ~isempty( acorrp{i,1} ) && ~isnan( acorrp{i,1}(1,1) ) && strcmp( data{i,4},
'Small' ) && strcmp( data{i,3}, 'DFW' ) && strcmp( data{i,2}, 'APA' )

        foo1 = small_dfw_c_apa_p;
        foot = acorrp{i,1}(1,2);
        foo2 = acorrp{i,1}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        small_dfw_c_apa_p = foo1;

    end

    if ~isempty( acorrp{i,1} ) && ~isnan( acorrp{i,2}(1,1) ) && strcmp( data{i,4},
'Small' ) && strcmp( data{i,3}, 'DFW' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = small_dfw_c_tdp_p;
        foot = acorrp{i,1}(1,2);
```

```
        foo2 = acorrp{i,1}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        small_dfw_c_tdp_p = foo1;

    end

    % DFW - small - y
    if ~isempty( acorrp{i,2} ) && ~isnan( acorrp{i,2}(1,1) ) && strcmp( data{i,4},
'Small' ) && strcmp( data{i,3}, 'DFW' ) && strcmp( data{i,2}, 'APA' )

        foo1 = small_dfw_y_apa_p;
        foot = acorrp{i,2}(1,2);
        foo2 = acorrp{i,2}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        small_dfw_y_apa_p = foo1;

    end

    if ~isempty( acorrp{i,2} ) && ~isnan( acorrp{i,2}(1,1) ) && strcmp( data{i,4},
'Small' ) && strcmp( data{i,3}, 'DFW' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = small_dfw_y_tdp_p;
        foot = acorrp{i,2}(1,2);
        foo2 = acorrp{i,2}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        small_dfw_y_tdp_p = foo1;

    end

    % DFW - small - z
    if ~isempty( acorrp{i,3} ) && ~isnan( acorrp{i,3}(1,1) ) && strcmp( data{i,4},
'Small' ) && strcmp( data{i,3}, 'DFW' ) && strcmp( data{i,2}, 'APA' )

        foo1 = small_dfw_z_apa_p;
        foot = acorrp{i,3}(1,2);
        foo2 = acorrp{i,3}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        small_dfw_z_apa_p = foo1;

    end

    if ~isempty( acorrp{i,3} ) && ~isnan( acorrp{i,3}(1,1) ) && strcmp( data{i,4},
'Small' ) && strcmp( data{i,3}, 'DFW' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = small_dfw_z_tdp_p;
        foot = acorrp{i,3}(1,2);
        foo2 = acorrp{i,3}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        small_dfw_z_tdp_p = foo1;
```

```
    end

    % MEM - small - circulation
    if ~isempty( acorrp{i,1} ) && ~isnan( acorrp{i,1}(1,1) ) && strcmp( data{i,4},
'Small' ) && strcmp( data{i,3}, 'MEM' ) && strcmp( data{i,2}, 'APA' )

        foo1 = small_mem_c_apa_p;
        foot = acorrp{i,1}(1,2);
        foo2 = acorrp{i,1}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        small_mem_c_apa_p = foo1;

    end

    if ~isempty( acorrp{i,1} ) && ~isnan( acorrp{i,1}(1,1) ) && strcmp( data{i,4},
'Small' ) && strcmp( data{i,3}, 'MEM' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = small_mem_c_tdp_p;
        foot = acorrp{i,1}(1,2);
        foo2 = acorrp{i,1}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        small_mem_c_tdp_p = foo1;

    end

    % MEM acorruals - small - y
    if ~isempty( acorrp{i,2} ) && ~isnan( acorrp{i,2}(1,1) ) && strcmp( data{i,4},
'Small' ) && strcmp( data{i,3}, 'MEM' ) && strcmp( data{i,2}, 'APA' )

        foo1 = small_mem_y_apa_p;
        foot = acorrp{i,2}(1,2);
        foo2 = acorrp{i,2}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        small_mem_y_apa_p = foo1;

    end

    if ~isempty( acorrp{i,2} ) && ~isnan( acorrp{i,2}(1,1) ) && strcmp( data{i,4},
'Small' ) && strcmp( data{i,3}, 'MEM' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = small_mem_y_tdp_p;
        foot = acorrp{i,2}(1,2);
        foo2 = acorrp{i,2}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        small_mem_y_tdp_p = foo1;

    end

    % MEM - small - z
    if ~isempty( acorrp{i,3} ) && ~isnan( acorrp{i,3}(1,1) ) && strcmp( data{i,4},
'Small' ) && strcmp( data{i,3}, 'MEM' ) && strcmp( data{i,2}, 'APA' )

        foo1 = small_mem_z_apa_p;
        foot = acorrp{i,3}(1,2);
```

```
        foo2 = acorrp{i,3}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        small_mem_z_apa_p = foo1;

    end

    if ~isempty( acorrp{i,3} ) && ~isnan( acorrp{i,3}(1,1) ) && strcmp( data{i,4},
'Small' ) && strcmp( data{i,3}, 'MEM' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = small_mem_z_tdp_p;
        foot = acorrp{i,3}(1,2);
        foo2 = acorrp{i,3}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        small_mem_z_tdp_p = foo1;

    end

    % DFW - large - circulation
    if ~isempty( acorrp{i,1} ) && ~isnan( acorrp{i,1}(1,1) ) && strcmp( data{i,4},
'Large' ) && strcmp( data{i,3}, 'DFW' ) && strcmp( data{i,2}, 'APA' )

        foo1 = large_dfw_c_apa_p;
        foot = acorrp{i,1}(1,2);
        foo2 = acorrp{i,1}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        large_dfw_c_apa_p = foo1;

    end

    if ~isempty( acorrp{i,1} ) && ~isnan( acorrp{i,1}(1,1) ) && strcmp( data{i,4},
'Large' ) && strcmp( data{i,3}, 'DFW' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = large_dfw_c_tdp_p;
        foot = acorrp{i,1}(1,2);
        foo2 = acorrp{i,1}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        large_dfw_c_tdp_p = foo1;

    end

    % DFW - large - y
    if ~isempty( acorrp{i,2} ) && ~isnan( acorrp{i,2}(1,1) ) && strcmp( data{i,4},
'Large' ) && strcmp( data{i,3}, 'DFW' ) && strcmp( data{i,2}, 'APA' )

        foo1 = large_dfw_y_apa_p;
        foot = acorrp{i,2}(1,2);
        foo2 = acorrp{i,2}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        large_dfw_y_apa_p = foo1;
```

```
    end

    if ~isempty( acorrp{i,2} ) && ~isnan( acorrp{i,2}(1,1) ) && strcmp( data{i,4},
'Large' ) && strcmp( data{i,3}, 'DFW' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = large_dfw_y_tdp_p;
        foot = acorrp{i,2}(1,2);
        foo2 = acorrp{i,2}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        large_dfw_y_tdp_p = foo1;

    end

    % DFW - large - z
    if ~isempty( acorrp{i,3} ) && ~isnan( acorrp{i,3}(1,1) ) && strcmp( data{i,4},
'Large' ) && strcmp( data{i,3}, 'DFW' ) && strcmp( data{i,2}, 'APA' )

        foo1 = large_dfw_z_apa_p;
        foot = acorrp{i,3}(1,2);
        foo2 = acorrp{i,3}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        large_dfw_z_apa_p = foo1;

    end

    if ~isempty( acorrp{i,3} ) && ~isnan( acorrp{i,3}(1,1) ) && strcmp( data{i,4},
'Large' ) && strcmp( data{i,3}, 'DFW' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = large_dfw_z_tdp_p;
        foot = acorrp{i,3}(1,2);
        foo2 = acorrp{i,3}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        large_dfw_z_tdp_p = foo1;

    end

    % MEM - large - circulation
    if ~isempty( acorrp{i,1} ) && ~isnan( acorrp{i,1}(1,1) ) && strcmp( data{i,4},
'Large' ) && strcmp( data{i,3}, 'MEM' ) && strcmp( data{i,2}, 'APA' )

        foo1 = large_mem_c_apa_p;
        foot = acorrp{i,1}(1,2);
        foo2 = acorrp{i,1}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        large_mem_c_apa_p = foo1;

    end

    if ~isempty( acorrp{i,1} ) && ~isnan( acorrp{i,1}(1,1) ) && strcmp( data{i,4},
'Large' ) && strcmp( data{i,3}, 'MEM' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = large_mem_c_tdp_p;
        foot = acorrp{i,1}(1,2);
        foo2 = acorrp{i,1}(:,1);
```

```
        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        large_mem_c_tdp_p = foo1;

    end

    % MEM - large - y
    if ~isempty( acorrp{i,2} ) && ~isnan( acorrp{i,2}(1,1) ) && strcmp( data{i,4},
'Large' ) && strcmp( data{i,3}, 'MEM' ) && strcmp( data{i,2}, 'APA' )

        foo1 = large_mem_y_apa_p;
        foot = acorrp{i,2}(1,2);
        foo2 = acorrp{i,2}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        large_mem_y_apa_p = foo1;

    end

    if ~isempty( acorrp{i,2} ) && ~isnan( acorrp{i,2}(1,1) ) && strcmp( data{i,4},
'Large' ) && strcmp( data{i,3}, 'MEM' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = large_mem_y_tdp_p;
        foot = acorrp{i,2}(1,2);
        foo2 = acorrp{i,2}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        large_mem_y_tdp_p = foo1;

    end

    % MEM - large - z
    if ~isempty( acorrp{i,3} ) && ~isnan( acorrp{i,3}(1,1) ) && strcmp( data{i,4},
'Large' ) && strcmp( data{i,3}, 'MEM' ) && strcmp( data{i,2}, 'APA' )

        foo1 = large_mem_z_apa_p;
        foot = acorrp{i,3}(1,2);
        foo2 = acorrp{i,3}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        large_mem_z_apa_p = foo1;

    end

    if ~isempty( acorrp{i,3} ) && ~isnan( acorrp{i,3}(1,1) ) && strcmp( data{i,4},
'Large' ) && strcmp( data{i,3}, 'MEM' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = large_mem_z_tdp_p;
        foot = acorrp{i,3}(1,2);
        foo2 = acorrp{i,3}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        large_mem_z_tdp_p = foo1;

    end
```

```
    % DFW - heavy - circulation
    if ~isempty( acorrp{i,1} ) && ~isnan( acorrp{i,1}(1,1) ) && strcmp( data{i,4},
'Heavy' ) && strcmp( data{i,3}, 'DFW' ) && strcmp( data{i,2}, 'APA' )

        foo1 = heavy_dfw_c_apa_p;
        foot = acorrp{i,1}(1,2);
        foo2 = acorrp{i,1}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        heavy_dfw_c_apa_p = foo1;

    end

    if ~isempty( acorrp{i,1} ) && ~isnan( acorrp{i,1}(1,1) ) && strcmp( data{i,4},
'Heavy' ) && strcmp( data{i,3}, 'DFW' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = heavy_dfw_c_tdp_p;
        foot = acorrp{i,1}(1,2);
        foo2 = acorrp{i,1}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        heavy_dfw_c_tdp_p = foo1;

    end

    % DFW - heavy - y
    if ~isempty( acorrp{i,2} ) && ~isnan( acorrp{i,2}(1,1) ) && strcmp( data{i,4},
'Heavy' ) && strcmp( data{i,3}, 'DFW' ) && strcmp( data{i,2}, 'APA' )

        foo1 = heavy_dfw_y_apa_p;
        foot = acorrp{i,2}(1,2);
        foo2 = acorrp{i,2}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        heavy_dfw_y_apa_p = foo1;

    end

    if ~isempty( acorrp{i,2} ) && ~isnan( acorrp{i,2}(1,1) ) && strcmp( data{i,4},
'Heavy' ) && strcmp( data{i,3}, 'DFW' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = heavy_dfw_y_tdp_p;
        foot = acorrp{i,2}(1,2);
        foo2 = acorrp{i,2}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        heavy_dfw_y_tdp_p = foo1;

    end

    % DFW - heavy - z
    if ~isempty( acorrp{i,3} ) && ~isnan( acorrp{i,3}(1,1) ) && strcmp( data{i,4},
'Heavy' ) && strcmp( data{i,3}, 'DFW' ) && strcmp( data{i,2}, 'APA' )

        foo1 = heavy_dfw_z_apa_p;
        foot = acorrp{i,3}(1,2);
        foo2 = acorrp{i,3}(:,1);
```

```
        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        heavy_dfw_z_apa_p = foo1;

    end

    if ~isempty( acorrp{i,3} ) && ~isnan( acorrp{i,3}(1,1) ) && strcmp( data{i,4},
'Heavy' ) && strcmp( data{i,3}, 'DFW' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = heavy_dfw_z_tdp_p;
        foot = acorrp{i,3}(1,2);
        foo2 = acorrp{i,3}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        heavy_dfw_z_tdp_p = foo1;

    end

    % MEM - heavy - circulation
    if ~isempty( acorrp{i,1} ) && ~isnan( acorrp{i,1}(1,1) ) && strcmp( data{i,4},
'Heavy' ) && strcmp( data{i,3}, 'MEM' ) && strcmp( data{i,2}, 'APA' )

        foo1 = heavy_mem_c_apa_p;
        foot = acorrp{i,1}(1,2);
        foo2 = acorrp{i,1}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        heavy_mem_c_apa_p = foo1;

    end

    if ~isempty( acorrp{i,1} ) && ~isnan( acorrp{i,1}(1,1) ) && strcmp( data{i,4},
'Heavy' ) && strcmp( data{i,3}, 'MEM' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = heavy_mem_c_tdp_p;
        foot = acorrp{i,1}(1,2);
        foo2 = acorrp{i,1}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        heavy_mem_c_tdp_p = foo1;

    end

    % MEM - heavy - y
    if ~isempty( acorrp{i,2} ) && ~isnan( acorrp{i,2}(1,1) ) && strcmp( data{i,4},
'Heavy' ) && strcmp( data{i,3}, 'MEM' ) && strcmp( data{i,2}, 'APA' )

        foo1 = heavy_mem_y_apa_p;
        foot = acorrp{i,2}(1,2);
        foo2 = acorrp{i,2}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        heavy_mem_y_apa_p = foo1;

    end
```

```
    if ~isempty( acorrp{i,2} ) && ~isnan( acorrp{i,2}(1,1) ) && strcmp( data{i,4},
'Heavy' ) && strcmp( data{i,3}, 'MEM' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = heavy_mem_y_tdp_p;
        foot = acorrp{i,2}(1,2);
        foo2 = acorrp{i,2}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        heavy_mem_y_tdp_p = foo1;

    end

    % MEM - heavy - z
    if ~isempty( acorrp{i,3} ) && ~isnan( acorrp{i,3}(1,1) ) && strcmp( data{i,4},
'Heavy' ) && strcmp( data{i,3}, 'MEM' ) && strcmp( data{i,2}, 'APA' )

        foo1 = heavy_mem_z_apa_p;
        foot = acorrp{i,3}(1,2);
        foo2 = acorrp{i,3}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        heavy_mem_z_apa_p = foo1;

    end

    if ~isempty( acorrp{i,3} ) && ~isnan( acorrp{i,3}(1,1) ) && strcmp( data{i,4},
'Heavy' ) && strcmp( data{i,3}, 'MEM' ) && strcmp( data{i,2}, 'TDP' )

        foo1 = heavy_mem_z_tdp_p;
        foot = acorrp{i,3}(1,2);
        foo2 = acorrp{i,3}(:,1);

        [rows,cols] = size( foo2 );
        ind = find( foo1(:,1) == foot );
        foo1 = [foo1, [NaN*ones(ind,1); foo2; NaN*ones(401-ind-rows,1)] ];

        heavy_mem_z_tdp_p = foo1;

    end

end

% Find Min, Mean, Max across all data sets at each lag time
varlabelsp = {'small_dfw_c_apa_p', 'small_dfw_c_tdp_p', 'small_dfw_y_apa_p',
'small_dfw_y_tdp_p', 'small_dfw_z_apa_p', 'small_dfw_z_tdp_p', ...
              'small_mem_c_apa_p', 'small_mem_c_tdp_p', 'small_mem_y_apa_p',
'small_mem_y_tdp_p', 'small_mem_z_apa_p', 'small_mem_z_tdp_p', ...
              'large_dfw_c_apa_p', 'large_dfw_c_tdp_p', 'large_dfw_y_apa_p',
'large_dfw_y_tdp_p', 'large_dfw_z_apa_p', 'large_dfw_z_tdp_p', ...
              'large_mem_c_apa_p', 'large_mem_c_tdp_p', 'large_mem_y_apa_p',
'large_mem_y_tdp_p', 'large_mem_z_apa_p', 'large_mem_z_tdp_p', ...
              'heavy_dfw_c_apa_p', 'heavy_dfw_c_tdp_p', 'heavy_dfw_y_apa_p',
'heavy_dfw_y_tdp_p', 'heavy_dfw_z_apa_p', 'heavy_dfw_z_tdp_p', ...
              'heavy_mem_c_apa_p', 'heavy_mem_c_tdp_p', 'heavy_mem_y_apa_p',
'heavy_mem_y_tdp_p', 'heavy_mem_z_apa_p', 'heavy_mem_z_tdp_p' };

for i = 1:length( varlabelsp )

    eval( ['mmm_', varlabelsp{i}, '(:,2) = nanmean( ', varlabelsp{i}, '(:,2:end), 2
);'] );
    eval( ['mmm_', varlabelsp{i}, '(:,4) =  nanstd( ', varlabelsp{i}, '(:,2:end), 0, 2
);'] );
```

```matlab
    eval( ['mmm_', varlabelsp{i}, '(:,1) =  mmm_', varlabelsp{i}, '(:,2) - 3*mmm_',
varlabelsp{i}, '(:,4);'] );
    eval( ['mmm_', varlabelsp{i}, '(:,3) =  mmm_', varlabelsp{i}, '(:,2) + 3*mmm_',
varlabelsp{i}, '(:,4);'] );

end


% Plot Results
% SMALL -----------------------------------------------------------------
figure( 'Name', 'Small - C' )
subplot( 1, 2, 1 )
plot( (foo1(:,1)-1)*speedms, mmm_small_dfw_c_apa_p(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_small_dfw_c_tdp_p(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_small_dfw_c_apa_p(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_small_dfw_c_apa_p(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_small_dfw_c_tdp_p(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_small_dfw_c_tdp_p(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Circulation Cross-Correlation, DFW'; '1m vs. 100m Predicted'} )
axis( [0 200 0 2] )

subplot( 1, 2, 2 )
plot( (foo1(:,1)-1)*speedms, mmm_small_mem_c_apa_p(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_small_mem_c_tdp_p(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_small_mem_c_apa_p(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_small_mem_c_apa_p(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_small_mem_c_tdp_p(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_small_mem_c_tdp_p(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Circulation Cross-Correlation, MEM'; '1m vs. 100m Predicted'} )
axis( [0 200 0 2] )


figure( 'Name', 'Small - Y' )
subplot( 1, 2, 1 )
plot( (foo1(:,1)-1)*speedms, mmm_small_dfw_y_apa_p(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_small_dfw_y_tdp_p(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_small_dfw_y_apa_p(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_small_dfw_y_apa_p(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_small_dfw_y_tdp_p(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_small_dfw_y_tdp_p(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Y-Pos. Cross-Correlation, DFW'; '1m vs. 100m Predicted'} )
axis( [0 200 0 2] )

subplot( 1, 2, 2 )
plot( (foo1(:,1)-1)*speedms, mmm_small_mem_y_apa_p(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_small_mem_y_tdp_p(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
```

```
plot( (foo1(:,1)-1)*speedms, mmm_small_mem_y_apa_p(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_small_mem_y_apa_p(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_small_mem_y_tdp_p(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_small_mem_y_tdp_p(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Y-Pos. Cross-Correlation, MEM'; '1m vs. 100m Predicted'} )
axis( [0 200 0 2] )


figure( 'Name', 'Small - Z' )
subplot( 1, 2, 1 )
plot( (foo1(:,1)-1)*speedms, mmm_small_dfw_z_apa_p(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_small_dfw_z_tdp_p(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_small_dfw_z_apa_p(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_small_dfw_z_apa_p(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_small_dfw_z_tdp_p(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_small_dfw_z_tdp_p(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Z-Pos. Cross-Correlation, DFW'; '1m vs. 100m Predicted'} )
axis( [0 200 0 2] )

subplot( 1, 2, 2 )
plot( (foo1(:,1)-1)*speedms, mmm_small_mem_z_apa_p(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_small_mem_z_tdp_p(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_small_mem_z_apa_p(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_small_mem_z_apa_p(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_small_mem_z_tdp_p(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_small_mem_z_tdp_p(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Z-Pos. Cross-Correlation, MEM'; '1m vs. 100m Predicted'} )
axis( [0 200 0 2] )


% LARGE ----------------------------------------------------------------
figure( 'Name', 'large - C' )
subplot( 1, 2, 1 )
plot( (foo1(:,1)-1)*speedms, mmm_large_dfw_c_apa_p(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_large_dfw_c_tdp_p(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_large_dfw_c_apa_p(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_large_dfw_c_apa_p(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_large_dfw_c_tdp_p(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_large_dfw_c_tdp_p(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Circulation Cross-Correlation, DFW'; '1m vs. 100m Predicted'} )
axis( [0 200 0 2] )

subplot( 1, 2, 2 )
plot( (foo1(:,1)-1)*speedms, mmm_large_mem_c_apa_p(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_large_mem_c_tdp_p(:,2), 'b.-', ...
```

```
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_large_mem_c_apa_p(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_large_mem_c_apa_p(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_large_mem_c_tdp_p(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_large_mem_c_tdp_p(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Circulation Cross-Correlation, MEM'; '1m vs. 100m Predicted'} )
axis( [0 200 0 2] )


figure( 'Name', 'large - Y' )
subplot( 1, 2, 1 )
plot( (foo1(:,1)-1)*speedms, mmm_large_dfw_y_apa_p(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_large_dfw_y_tdp_p(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_large_dfw_y_apa_p(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_large_dfw_y_apa_p(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_large_dfw_y_tdp_p(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_large_dfw_y_tdp_p(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Y-Pos. Cross-Correlation, DFW'; '1m vs. 100m Predicted'} )
axis( [0 200 0 2] )

subplot( 1, 2, 2 )
plot( (foo1(:,1)-1)*speedms, mmm_large_mem_y_apa_p(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_large_mem_y_tdp_p(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_large_mem_y_apa_p(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_large_mem_y_apa_p(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_large_mem_y_tdp_p(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_large_mem_y_tdp_p(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Y-Pos. Cross-Correlation, MEM'; '1m vs. 100m Predicted'} )
axis( [0 200 0 2] )


figure( 'Name', 'large - Z' )
subplot( 1, 2, 1 )
plot( (foo1(:,1)-1)*speedms, mmm_large_dfw_z_apa_p(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_large_dfw_z_tdp_p(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_large_dfw_z_apa_p(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_large_dfw_z_apa_p(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_large_dfw_z_tdp_p(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_large_dfw_z_tdp_p(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Z-Pos. Cross-Correlation, DFW'; '1m vs. 100m Predicted'} )
axis( [0 200 0 2] )

subplot( 1, 2, 2 )
```

```
plot( (foo1(:,1)-1)*speedms, mmm_large_mem_z_apa_p(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_large_mem_z_tdp_p(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_large_mem_z_apa_p(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_large_mem_z_apa_p(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_large_mem_z_tdp_p(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_large_mem_z_tdp_p(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Z-Pos. Cross-Correlation, MEM'; '1m vs. 100m Predicted'} )
axis( [0 200 0 2] )


% HEAVY ------------------------------------------------------------------
figure( 'Name', 'heavy - C' )
subplot( 1, 2, 1 )
plot( (foo1(:,1)-1)*speedms, mmm_heavy_dfw_c_apa_p(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_dfw_c_tdp_p(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_heavy_dfw_c_apa_p(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_dfw_c_apa_p(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_dfw_c_tdp_p(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_dfw_c_tdp_p(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Circulation Cross-Correlation, DFW'; '1m vs. 100m Predicted'} )
axis( [0 200 0 2] )

subplot( 1, 2, 2 )
plot( (foo1(:,1)-1)*speedms, mmm_heavy_mem_c_apa_p(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_mem_c_tdp_p(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_heavy_mem_c_apa_p(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_mem_c_apa_p(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_mem_c_tdp_p(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_mem_c_tdp_p(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Circulation Cross-Correlation, MEM'; '1m vs. 100m Predicted'} )
axis( [0 200 0 2] )


figure( 'Name', 'heavy - Y' )
subplot( 1, 2, 1 )
plot( (foo1(:,1)-1)*speedms, mmm_heavy_dfw_y_apa_p(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_dfw_y_tdp_p(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_heavy_dfw_y_apa_p(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_dfw_y_apa_p(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_dfw_y_tdp_p(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_dfw_y_tdp_p(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Y-Pos. Cross-Correlation, DFW'; '1m vs. 100m Predicted'} )
```

```
axis( [0 200 0 2] )

subplot( 1, 2, 2 )
plot( (foo1(:,1)-1)*speedms, mmm_heavy_mem_y_apa_p(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_mem_y_tdp_p(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_heavy_mem_y_apa_p(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_mem_y_apa_p(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_mem_y_tdp_p(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_mem_y_tdp_p(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Y-Pos. Cross-Correlation, MEM'; '1m vs. 100m Predicted'} )
axis( [0 200 0 2] )


figure( 'Name', 'heavy - Z' )
subplot( 1, 2, 1 )
plot( (foo1(:,1)-1)*speedms, mmm_heavy_dfw_z_apa_p(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_dfw_z_tdp_p(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_heavy_dfw_z_apa_p(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_dfw_z_apa_p(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_dfw_z_tdp_p(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_dfw_z_tdp_p(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Z-Pos. Cross-Correlation, DFW'; '1m vs. 100m Predicted'} )
axis( [0 200 0 2] )

subplot( 1, 2, 2 )
plot( (foo1(:,1)-1)*speedms, mmm_heavy_mem_z_apa_p(:,2), 'r.-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_mem_z_tdp_p(:,2), 'b.-', ...
      'linewidth', 2 )
legend( 'APA', 'TDP' )
hold on
plot( (foo1(:,1)-1)*speedms, mmm_heavy_mem_z_apa_p(:,1), 'rv-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_mem_z_apa_p(:,3), 'r^-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_mem_z_tdp_p(:,1), 'bv-', ...
      (foo1(:,1)-1)*speedms, mmm_heavy_mem_z_tdp_p(:,3), 'b^-', ...
      'linewidth', 2 )
grid on
xlabel( xstring )
ylabel( 'mean-3\sigma, mean, mean+3\sigma' )
title( {'Z-Pos. Cross-Correlation, MEM'; '1m vs. 100m Predicted'} )
axis( [0 200 0 2] )




% xlabel( 'Lag Time, l (s)' )
% ylabel( 'Cross-correlation' )
% title( 'Circulation Cross-correlation' )
% grid on
% axis( [0 180 0 1] )
%
% % Plot Cross-correlation - small - y
% subplot( 3, 2, 3 )
% for i = 1:n:imax
%
%     if ~isempty( acorrp{i,2} ) && strcmp( data{i,4}, 'Small' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'APA' )
```

```
%
%        plot( acorrp{i,2}(:,2), acorrp{i,2}(:,1), 'r.-' )
%        hold on
%
%     end
%
%     if ~isempty( acorrp{i,2} ) && strcmp( data{i,4}, 'Small' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'TDP' )
%
%        plot( acorrp{i,2}(:,2), acorrp{i,2}(:,1), 'b.-' )
%        hold on
%
%     end
%
% end
% xlabel( 'Lag Time, l (s)' )
% ylabel( 'Cross-correlation' )
% title( 'Y-Position Cross-correlation' )
% grid on
% axis( [0 180 0 1] )
%
% % Plot Cross-correlation - small - z
% subplot( 3, 2, 5 )
% for i = 1:n:imax
%
%     if ~isempty( acorrp{i,3} ) && strcmp( data{i,4}, 'Small' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'APA' )
%
%        plot( acorrp{i,3}(:,2), acorrp{i,3}(:,1), 'r.-' )
%        hold on
%
%     end
%
%     if ~isempty( acorrp{i,3} ) && strcmp( data{i,4}, 'Small' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'TDP' )
%
%        plot( acorrp{i,3}(:,2), acorrp{i,3}(:,1), 'b.-' )
%        hold on
%
%     end
%
% end
% xlabel( 'Lag Time, l (s)' )
% ylabel( 'Cross-correlation' )
% title( 'Z-Position Cross-correlation' )
% grid on
% axis( [0 180 0 1] )
%
% %%% SMALL - MEM
% % Plot Cross-correlation - small - circ
% subplot( 3, 2, 2 )
% for i = 1:n:imax
%
%     if ~isempty( acorrp{i,1} ) && strcmp( data{i,4}, 'Small' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'APA' )
%
%        plot( acorrp{i,1}(:,2), acorrp{i,1}(:,1), 'r.-' )
%        hold on
%
%     end
%
%     if ~isempty( acorrp{i,1} ) && strcmp( data{i,4}, 'Small' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'TDP' )
%
%        plot( acorrp{i,1}(:,2), acorrp{i,1}(:,1), 'b.-' )
%        hold on
%
%     end
%
% end
```

```
% xlabel( 'Lag Time, l (s)' )
% ylabel( 'Cross-correlation' )
% title( 'Circulation Cross-correlation' )
% grid on
% axis( [0 180 0 1] )
%
% % Plot Cross-correlation - small - y
% subplot( 3, 2, 4 )
% for i = 1:n:imax
%
%     if ~isempty( acorrp{i,2} ) && strcmp( data{i,4}, 'Small' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'APA' )
%
%        plot( acorrp{i,2}(:,2), acorrp{i,2}(:,1), 'r.-' )
%        hold on
%
%     end
%
%     if ~isempty( acorrp{i,2} ) && strcmp( data{i,4}, 'Small' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'TDP' )
%
%        plot( acorrp{i,2}(:,2), acorrp{i,2}(:,1), 'b.-' )
%        hold on
%
%     end
%
% end
% xlabel( 'Lag Time, l (s)' )
% ylabel( 'Cross-correlation' )
% title( 'Y-Position Cross-correlation' )
% grid on
% axis( [0 180 0 1] )
%
% % Plot Cross-correlation - small - z
% subplot( 3, 2, 6 )
% for i = 1:n:imax
%
%     if ~isempty( acorrp{i,3} ) && strcmp( data{i,4}, 'Small' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'APA' )
%
%        plot( acorrp{i,3}(:,2), acorrp{i,3}(:,1), 'r.-' )
%        hold on
%
%     end
%
%     if ~isempty( acorrp{i,3} ) && strcmp( data{i,4}, 'Small' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'TDP' )
%
%        plot( acorrp{i,3}(:,2), acorrp{i,3}(:,1), 'b.-' )
%        hold on
%
%     end
%
% end
% xlabel( 'Lag Time, l (s)' )
% ylabel( 'Cross-correlation' )
% title( 'Z-Position Cross-correlation' )
% grid on
% axis( [0 180 0 1] )
%
%
% %%% LARGE - DFW
% figure( 'Name', 'Large' );
% % Plot Cross-correlation - large - circ
% subplot( 3, 2, 1 )
% for i = 1:n:imax
%
%     if ~isempty( acorrp{i,1} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'APA' )
%
```

```
%        plot( acorrp{i,1}(:,2), acorrp{i,1}(:,1), 'r.-' )
%        hold on
%
%     end
%
%     if ~isempty( acorrp{i,1} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'TDP' )
%
%        plot( acorrp{i,1}(:,2), acorrp{i,1}(:,1), 'b.-' )
%        hold on
%
%     end
%
% end
% xlabel( 'Lag Time, l (s)' )
% ylabel( 'R_{x_{1}x_{m}}' )
% title( {'Circulation Cross-correlation'; 'Large Aircraft, DFW'} )
% grid on
% axis( [0 150 0 1] )
%
% % Plot Cross-correlation - large - y
% subplot( 3, 2, 3 )
% for i = 1:n:imax
%
%     if ~isempty( acorrp{i,2} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'APA' )
%
%        plot( acorrp{i,2}(:,2), acorrp{i,2}(:,1), 'r.-' )
%        hold on
%
%     end
%
%     if ~isempty( acorrp{i,2} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'TDP' )
%
%        plot( acorrp{i,2}(:,2), acorrp{i,2}(:,1), 'b.-' )
%        hold on
%
%     end
%
% end
% xlabel( 'Lag Time, l (s)' )
% ylabel( 'R_{x_{1}x_{m}}' )
% title( {'Y-Position Cross-correlation'; 'Large Aircraft, DFW'} )
% grid on
% axis( [0 150 0 1] )
%
% % Plot Cross-correlation - large - z
% subplot( 3, 2, 5 )
% for i = 1:n:imax
%
%     if ~isempty( acorrp{i,3} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'APA' )
%
%        plot( acorrp{i,3}(:,2), acorrp{i,3}(:,1), 'r.-' )
%        hold on
%
%     end
%
%     if ~isempty( acorrp{i,3} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'TDP' )
%
%        plot( acorrp{i,3}(:,2), acorrp{i,3}(:,1), 'b.-' )
%        hold on
%
%     end
%
% end
% xlabel( 'Lag Time, l (s)' )
% ylabel( 'R_{x_{1}x_{m}}' )
```

```
% title( {'Z-Position Cross-correlation'; 'Large Aircraft, DFW'} )
% grid on
% axis( [0 150 0 1] )
%
% %%% LARGE - MEM
% % Plot Cross-correlation - large - circ
% subplot( 3, 2, 2 )
% for i = 1:n:imax
%
%     if ~isempty( acorrp{i,1} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'APA' )
%
%         plot( acorrp{i,1}(:,2), acorrp{i,1}(:,1), 'r.-' )
%         hold on
%
%     end
%
%     if ~isempty( acorrp{i,1} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'TDP' )
%
%         plot( acorrp{i,1}(:,2), acorrp{i,1}(:,1), 'b.-' )
%         hold on
%
%     end
%
% end
% xlabel( 'Lag Time, l (s)' )
% ylabel( 'R_{x_{1}x_{m}}' )
% title( {'Circulation Cross-correlation'; 'Large Aircraft, MEM'} )
% grid on
% axis( [0 150 0 1] )
%
% % Plot Cross-correlation - large - y
% subplot( 3, 2, 4 )
% for i = 1:n:imax
%
%     if ~isempty( acorrp{i,2} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'APA' )
%
%         plot( acorrp{i,2}(:,2), acorrp{i,2}(:,1), 'r.-' )
%         hold on
%
%     end
%
%     if ~isempty( acorrp{i,2} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'TDP' )
%
%         plot( acorrp{i,2}(:,2), acorrp{i,2}(:,1), 'b.-' )
%         hold on
%
%     end
%
% end
% xlabel( 'Lag Time, l (s)' )
% ylabel( 'R_{x_{1}x_{m}}' )
% title( {'Y-Position Cross-correlation'; 'Large Aircraft, MEM'} )
% grid on
% axis( [0 150 0 1] )
%
% % Plot Cross-correlation - large - z
% subplot( 3, 2, 6 )
% for i = 1:n:imax
%
%     if ~isempty( acorrp{i,3} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'APA' )
%
%         plot( acorrp{i,3}(:,2), acorrp{i,3}(:,1), 'r.-' )
%         hold on
%
%     end
```

```
%
%     if ~isempty( acorrp{i,3} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'TDP' )
%
%         plot( acorrp{i,3}(:,2), acorrp{i,3}(:,1), 'b.-' )
%         hold on
%
%     end
%
% end
% xlabel( 'Lag Time, l (s)' )
% ylabel( 'R_{x_{1}x_{m}}' )
% title( {'Z-Position Cross-correlation'; 'Large Aircraft, MEM'} )
% grid on
% axis( [0 150 0 1] )
%
% %%% HEAVY - DFW
% figure( 'Name', 'Heavy' );
% % Plot Cross-correlation - heavy - circ
% subplot( 3, 2, 1 )
% for i = 1:n:imax
%
%     if ~isempty( acorrp{i,1} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'APA' )
%
%         plot( acorrp{i,1}(:,2), acorrp{i,1}(:,1), 'r.-' )
%         hold on
%
%     end
%
%     if ~isempty( acorrp{i,1} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'TDP' )
%
%         plot( acorrp{i,1}(:,2), acorrp{i,1}(:,1), 'b.-' )
%         hold on
%
%     end
%
% end
% xlabel( 'Lag Time, l (s)' )
% ylabel( 'R_{x_{1}x_{m}}' )
% title( {'Circulation Cross-correlation'; 'Heavy Aircraft, DFW'} )
% grid on
% axis( [0 150 0 1] )
%
% % Plot Cross-correlation - heavy - y
% subplot( 3, 2, 3 )
% for i = 1:n:imax
%
%     if ~isempty( acorrp{i,2} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'APA' )
%
%         plot( acorrp{i,2}(:,2), acorrp{i,2}(:,1), 'r.-' )
%         hold on
%
%     end
%
%     if ~isempty( acorrp{i,2} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'TDP' )
%
%         plot( acorrp{i,2}(:,2), acorrp{i,2}(:,1), 'b.-' )
%         hold on
%
%     end
%
% end
% xlabel( 'Lag Time, l (s)' )
% ylabel( 'R_{x_{1}x_{m}}' )
% title( {'Y-Position Cross-correlation'; 'Heavy Aircraft, DFW'} )
% grid on
```

```
% axis( [0 150 0 1] )
%
% % Plot Cross-correlation - heavy - z
% subplot( 3, 2, 5 )
% for i = 1:n:imax
%
%     if ~isempty( acorrp{i,3} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'APA' )
%
%         plot( acorrp{i,3}(:,2), acorrp{i,3}(:,1), 'r.-' )
%         hold on
%
%     end
%
%     if ~isempty( acorrp{i,3} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'TDP' )
%
%         plot( acorrp{i,3}(:,2), acorrp{i,3}(:,1), 'b.-' )
%         hold on
%
%     end
%
% end
% xlabel( 'Lag Time, l (s)' )
% ylabel( 'R_{x_{1}x_{m}}' )
% title( {'Z-Position Cross-correlation'; 'Heavy Aircraft, DFW'} )
% grid on
% axis( [0 150 0 1] )
%
% %%% HEAVY - MEM
% % Plot Cross-correlation - heavy - circ
% subplot( 3, 2, 2 )
% for i = 1:n:imax
%
%     if ~isempty( acorrp{i,1} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'APA' )
%
%         plot( acorrp{i,1}(:,2), acorrp{i,1}(:,1), 'r.-' )
%         hold on
%
%     end
%
%     if ~isempty( acorrp{i,1} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'TDP' )
%
%         plot( acorrp{i,1}(:,2), acorrp{i,1}(:,1), 'b.-' )
%         hold on
%
%     end
%
% end
% xlabel( 'Lag Time, l (s)' )
% ylabel( 'R_{x_{1}x_{m}}' )
% title( {'Circulation Cross-correlation'; 'Heavy Aircraft, MEM'} )
% grid on
% axis( [0 150 0 1] )
%
% % Plot Cross-correlation - heavy - y
% subplot( 3, 2, 4 )
% for i = 1:n:imax
%
%     if ~isempty( acorrp{i,2} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'APA' )
%
%         plot( acorrp{i,2}(:,2), acorrp{i,2}(:,1), 'r.-' )
%         hold on
%
%     end
%
```

```
%     if ~isempty( acorrp{i,2} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'TDP' )
%
%        plot( acorrp{i,2}(:,2), acorrp{i,2}(:,1), 'b.-' )
%        hold on
%
%     end
%
% end
% xlabel( 'Lag Time, l (s)' )
% ylabel( 'R_{x_{1}x_{m}}' )
% title( {'Y-Position Cross-correlation'; 'Heavy Aircraft, MEM'} )
% grid on
% axis( [0 150 0 1] )
%
% % Plot Cross-correlation - heavy - z
% subplot( 3, 2, 6 )
% for i = 1:n:imax
%
%     if ~isempty( acorrp{i,3} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'APA' )
%
%        plot( acorrp{i,3}(:,2), acorrp{i,3}(:,1), 'r.-' )
%        hold on
%
%     end
%
%     if ~isempty( acorrp{i,3} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'TDP' )
%
%        plot( acorrp{i,3}(:,2), acorrp{i,3}(:,1), 'b.-' )
%        hold on
%
%     end
%
% end
% xlabel( 'Lag Time, l (s)' )
% ylabel( 'R_{x_{1}x_{m}}' )
% title( {'Z-Position Cross-correlation'; 'Heavy Aircraft, MEM'} )
% grid on
% axis( [0 150 0 1] )
%
%
% % Small aircraft not plotted because of lack of cases ---------------------
% %%% Measured vs. Measured 100 ---------------------------------------------
% % %%% SMALL - DFW
% % figure( 'Name', 'Small' );
% % % Plot Cross-correlation - small - circ
% % subplot( 3, 2, 1 )
% % for i = 1:n:imax
% %
% %     if ~isempty( acorrp{i,1} ) && strcmp( data{i,4}, 'Small' ) && strcmp(
data{i,3}, 'DFW' ) && strcmp( data{i,2}, 'APA' )
% %
% %        plot( acorrp{i,1}(:,2), acorrp{i,1}(:,1), 'r.-' )
% %        hold on
% %
% %     end
% %
% %     if ~isempty( acorrp{i,1} ) && strcmp( data{i,4}, 'Small' ) && strcmp(
data{i,3}, 'DFW' ) && strcmp( data{i,2}, 'TDP' )
% %
% %        plot( acorrp{i,1}(:,2), acorrp{i,1}(:,1), 'b.-' )
% %        hold on
% %
% %     end
% %
% % end
% % xlabel( 'Lag Time, l (s)' )
% % ylabel( 'Cross-correlation' )
```

```
% % title( 'Circulation Cross-correlation' )
% % grid on
% % axis( [0 180 0 1] )
% %
% % % Plot Cross-correlation - small - y
% % subplot( 3, 2, 3 )
% % for i = 1:n:imax
% %
% %     if ~isempty( acorrp{i,2} ) && strcmp( data{i,4}, 'Small' ) && strcmp(
data{i,3}, 'DFW' ) && strcmp( data{i,2}, 'APA' )
% %
% %        plot( acorrp{i,2}(:,2), acorrp{i,2}(:,1), 'r.-' )
% %        hold on
% %
% %     end
% %
% %     if ~isempty( acorrp{i,2} ) && strcmp( data{i,4}, 'Small' ) && strcmp(
data{i,3}, 'DFW' ) && strcmp( data{i,2}, 'TDP' )
% %
% %        plot( acorrp{i,2}(:,2), acorrp{i,2}(:,1), 'b.-' )
% %        hold on
% %
% %     end
% %
% % end
% % xlabel( 'Lag Time, l (s)' )
% % ylabel( 'Cross-correlation' )
% % title( 'Y-Position Cross-correlation' )
% % grid on
% % axis( [0 180 0 1] )
% %
% % % Plot Cross-correlation - small - z
% % subplot( 3, 2, 5 )
% % for i = 1:n:imax
% %
% %     if ~isempty( acorrp{i,3} ) && strcmp( data{i,4}, 'Small' ) && strcmp(
data{i,3}, 'DFW' ) && strcmp( data{i,2}, 'APA' )
% %
% %        plot( acorrp{i,3}(:,2), acorrp{i,3}(:,1), 'r.-' )
% %        hold on
% %
% %     end
% %
% %     if ~isempty( acorrp{i,3} ) && strcmp( data{i,4}, 'Small' ) && strcmp(
data{i,3}, 'DFW' ) && strcmp( data{i,2}, 'TDP' )
% %
% %        plot( acorrp{i,3}(:,2), acorrp{i,3}(:,1), 'b.-' )
% %        hold on
% %
% %     end
% %
% % end
% % xlabel( 'Lag Time, l (s)' )
% % ylabel( 'Cross-correlation' )
% % title( 'Z-Position Cross-correlation' )
% % grid on
% % axis( [0 180 0 1] )
% %
% % %%% SMALL - MEM
% % % Plot Cross-correlation - small - circ
% % subplot( 3, 2, 2 )
% % for i = 1:n:imax
% %
% %     if ~isempty( acorrp{i,1} ) && strcmp( data{i,4}, 'Small' ) && strcmp(
data{i,3}, 'MEM' ) && strcmp( data{i,2}, 'APA' )
% %
% %        plot( acorrp{i,1}(:,2), acorrp{i,1}(:,1), 'r.-' )
% %        hold on
% %
% %     end
```

```
% %
% %     if ~isempty( acorrp{i,1} ) && strcmp( data{i,4}, 'Small' ) && strcmp(
data{i,3}, 'MEM' ) && strcmp( data{i,2}, 'TDP' )
% %
% %         plot( acorrp{i,1}(:,2), acorrp{i,1}(:,1), 'b.-' )
% %         hold on
% %
% %     end
% %
% % end
% % xlabel( 'Lag Time, l (s)' )
% % ylabel( 'Cross-correlation' )
% % title( 'Circulation Cross-correlation' )
% % grid on
% % axis( [0 180 0 1] )
% %
% % % Plot Cross-correlation - small - y
% % subplot( 3, 2, 4 )
% % for i = 1:n:imax
% %
% %     if ~isempty( acorrp{i,2} ) && strcmp( data{i,4}, 'Small' ) && strcmp(
data{i,3}, 'MEM' ) && strcmp( data{i,2}, 'APA' )
% %
% %         plot( acorrp{i,2}(:,2), acorrp{i,2}(:,1), 'r.-' )
% %         hold on
% %
% %     end
% %
% %     if ~isempty( acorrp{i,2} ) && strcmp( data{i,4}, 'Small' ) && strcmp(
data{i,3}, 'MEM' ) && strcmp( data{i,2}, 'TDP' )
% %
% %         plot( acorrp{i,2}(:,2), acorrp{i,2}(:,1), 'b.-' )
% %         hold on
% %
% %     end
% %
% % end
% % xlabel( 'Lag Time, l (s)' )
% % ylabel( 'Cross-correlation' )
% % title( 'Y-Position Cross-correlation' )
% % grid on
% % axis( [0 180 0 1] )
% %
% % % Plot Cross-correlation - small - z
% % subplot( 3, 2, 6 )
% % for i = 1:n:imax
% %
% %     if ~isempty( acorrp{i,3} ) && strcmp( data{i,4}, 'Small' ) && strcmp(
data{i,3}, 'MEM' ) && strcmp( data{i,2}, 'APA' )
% %
% %         plot( acorrp{i,3}(:,2), acorrp{i,3}(:,1), 'r.-' )
% %         hold on
% %
% %     end
% %
% %     if ~isempty( acorrp{i,3} ) && strcmp( data{i,4}, 'Small' ) && strcmp(
data{i,3}, 'MEM' ) && strcmp( data{i,2}, 'TDP' )
% %
% %         plot( acorrp{i,3}(:,2), acorrp{i,3}(:,1), 'b.-' )
% %         hold on
% %
% %     end
% % end
% % xlabel( 'Lag Time, l (s)' )
% % ylabel( 'Cross-correlation' )
% % title( 'Z-Position Cross-correlation' )
% % grid on
% % axis( [0 180 0 1] )
%
%
```

```
% %%% LARGE - DFW
% figure( 'Name', 'Large' );
% % Plot Cross-correlation - large - circ
% subplot( 3, 2, 1 )
% for i = 1:n:imax
%
%     if ~isempty( acorrp{i,1} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'APA' )
%
%        plot( acorrp{i,1}(:,2), acorrp{i,1}(:,1), 'r.-' )
%        hold on
%
%     end
%
%     if ~isempty( acorrp{i,1} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'TDP' )
%
%        plot( acorrp{i,1}(:,2), acorrp{i,1}(:,1), 'b.-' )
%        hold on
%
%     end
%
% end
% xlabel( 'Lag Time, l (s)' )
% ylabel( 'R_{x_{1}x_{100}}' )
% title( {'Circulation Cross-correlation'; 'Large Aircraft, DFW'} )
% grid on
% axis( [0 180 0 1] )
%
% % Plot Cross-correlation - large - y
% subplot( 3, 2, 3 )
% for i = 1:n:imax
%
%     if ~isempty( acorrp{i,2} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'APA' )
%
%        plot( acorrp{i,2}(:,2), acorrp{i,2}(:,1), 'r.-' )
%        hold on
%
%     end
%
%     if ~isempty( acorrp{i,2} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'TDP' )
%
%        plot( acorrp{i,2}(:,2), acorrp{i,2}(:,1), 'b.-' )
%        hold on
%
%     end
%
% end
% xlabel( 'Lag Time, l (s)' )
% ylabel( 'R_{x_{1}x_{100}}' )
% title( {'Y-Position Cross-correlation'; 'Large Aircraft, DFW'} )
% grid on
% axis( [0 180 0 1] )
%
% % Plot Cross-correlation - large - z
% subplot( 3, 2, 5 )
% for i = 1:n:imax
%
%     if ~isempty( acorrp{i,3} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'APA' )
%
%        plot( acorrp{i,3}(:,2), acorrp{i,3}(:,1), 'r.-' )
%        hold on
%
%     end
%
%     if ~isempty( acorrp{i,3} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'TDP' )
```

```
%
%        plot( acorrp{i,3}(:,2), acorrp{i,3}(:,1), 'b.-' )
%        hold on
%
%     end
%
% end
% xlabel( 'Lag Time, l (s)' )
% ylabel( 'R_{x_{1}x_{100}}' )
% title( {'Z-Position Cross-correlation'; 'Large Aircraft, DFW'} )
% grid on
% axis( [0 180 0 1] )
%
% %%% LARGE - MEM
% % Plot Cross-correlation - large - circ
% subplot( 3, 2, 2 )
% for i = 1:n:imax
%
%     if ~isempty( acorrp{i,1} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'APA' )
%
%        plot( acorrp{i,1}(:,2), acorrp{i,1}(:,1), 'r.-' )
%        hold on
%
%     end
%
%     if ~isempty( acorrp{i,1} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'TDP' )
%
%        plot( acorrp{i,1}(:,2), acorrp{i,1}(:,1), 'b.-' )
%        hold on
%
%     end
%
% end
% xlabel( 'Lag Time, l (s)' )
% ylabel( 'R_{x_{1}x_{100}}' )
% title( {'Circulation Cross-correlation'; 'Large Aircraft, MEM'} )
% grid on
% axis( [0 180 0 1] )
%
% % Plot Cross-correlation - large - y
% subplot( 3, 2, 4 )
% for i = 1:n:imax
%
%     if ~isempty( acorrp{i,2} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'APA' )
%
%        plot( acorrp{i,2}(:,2), acorrp{i,2}(:,1), 'r.-' )
%        hold on
%
%     end
%
%     if ~isempty( acorrp{i,2} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'TDP' )
%
%        plot( acorrp{i,2}(:,2), acorrp{i,2}(:,1), 'b.-' )
%        hold on
%
%     end
%
% end
% xlabel( 'Lag Time, l (s)' )
% ylabel( 'R_{x_{1}x_{100}}' )
% title( {'Y-Position Cross-correlation'; 'Large Aircraft, MEM'} )
% grid on
% axis( [0 180 0 1] )
%
% % Plot Cross-correlation - large - z
% subplot( 3, 2, 6 )
```

```
% for i = 1:n:imax
%
%     if ~isempty( acorrp{i,3} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'APA' )
%
%         plot( acorrp{i,3}(:,2), acorrp{i,3}(:,1), 'r.-' )
%         hold on
%
%     end
%
%     if ~isempty( acorrp{i,3} ) && strcmp( data{i,4}, 'Large' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'TDP' )
%
%         plot( acorrp{i,3}(:,2), acorrp{i,3}(:,1), 'b.-' )
%         hold on
%
%     end
%
% end
% xlabel( 'Lag Time, l (s)' )
% ylabel( 'R_{x_{1}x_{100}}' )
% title( {'Z-Position Cross-correlation'; 'Large Aircraft, MEM'} )
% grid on
% axis( [0 180 0 1] )
%
% %%% HEAVY - DFW
% figure( 'Name', 'Heavy' );
% % Plot Cross-correlation - heavy - circ
% subplot( 3, 2, 1 )
% for i = 1:n:imax
%
%     if ~isempty( acorrp{i,1} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'APA' )
%
%         plot( acorrp{i,1}(:,2), acorrp{i,1}(:,1), 'r.-' )
%         hold on
%
%     end
%
%     if ~isempty( acorrp{i,1} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'TDP' )
%
%         plot( acorrp{i,1}(:,2), acorrp{i,1}(:,1), 'b.-' )
%         hold on
%
%     end
%
% end
% xlabel( 'Lag Time, l (s)' )
% ylabel( 'R_{x_{1}x_{100}}' )
% title( {'Circulation Cross-correlation'; 'Heavy Aircraft, DFW'} )
% grid on
% axis( [0 180 0 1] )
%
% % Plot Cross-correlation - heavy - y
% subplot( 3, 2, 3 )
% for i = 1:n:imax
%
%     if ~isempty( acorrp{i,2} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'APA' )
%
%         plot( acorrp{i,2}(:,2), acorrp{i,2}(:,1), 'r.-' )
%         hold on
%
%     end
%
%     if ~isempty( acorrp{i,2} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'TDP' )
%
%         plot( acorrp{i,2}(:,2), acorrp{i,2}(:,1), 'b.-' )
```

```
%         hold on
%
%     end
%
% end
% xlabel( 'Lag Time, l (s)' )
% ylabel( 'R_{x_{1}x_{100}}' )
% title( {'Y-Position Cross-correlation'; 'Heavy Aircraft, DFW'} )
% grid on
% axis( [0 180 0 1] )
%
% % Plot Cross-correlation - heavy - z
% subplot( 3, 2, 5 )
% for i = 1:n:imax
%
%     if ~isempty( acorrp{i,3} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'APA' )
%
%         plot( acorrp{i,3}(:,2), acorrp{i,3}(:,1), 'r.-' )
%         hold on
%
%     end
%
%     if ~isempty( acorrp{i,3} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'DFW' ) && strcmp( data{i,2}, 'TDP' )
%
%         plot( acorrp{i,3}(:,2), acorrp{i,3}(:,1), 'b.-' )
%         hold on
%
%     end
%
% end
% xlabel( 'Lag Time, l (s)' )
% ylabel( 'R_{x_{1}x_{100}}' )
% title( {'Z-Position Cross-correlation'; 'Heavy Aircraft, DFW'} )
% grid on
% axis( [0 180 0 1] )
%
% %%% HEAVY - MEM
% % Plot Cross-correlation - heavy - circ
% subplot( 3, 2, 2 )
% for i = 1:n:imax
%
%     if ~isempty( acorrp{i,1} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'APA' )
%
%         plot( acorrp{i,1}(:,2), acorrp{i,1}(:,1), 'r.-' )
%         hold on
%
%     end
%
%     if ~isempty( acorrp{i,1} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'TDP' )
%
%         plot( acorrp{i,1}(:,2), acorrp{i,1}(:,1), 'b.-' )
%         hold on
%
%     end
%
% end
% xlabel( 'Lag Time, l (s)' )
% ylabel( 'R_{x_{1}x_{100}}' )
% title( {'Circulation Cross-correlation'; 'Heavy Aircraft, MEM'} )
% grid on
% axis( [0 180 0 1] )
%
% % Plot Cross-correlation - heavy - y
% subplot( 3, 2, 4 )
% for i = 1:n:imax
%
```

```
%     if ~isempty( acorrp{i,2} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'APA' )
%
%        plot( acorrp{i,2}(:,2), acorrp{i,2}(:,1), 'r.-' )
%        hold on
%
%     end
%
%     if ~isempty( acorrp{i,2} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'TDP' )
%
%        plot( acorrp{i,2}(:,2), acorrp{i,2}(:,1), 'b.-' )
%        hold on
%
%     end
%
% end
% xlabel( 'Lag Time, l (s)' )
% ylabel( 'R_{x_{1}x_{100}}' )
% title( {'Y-Position Cross-correlation'; 'Heavy Aircraft, MEM'} )
% grid on
% axis( [0 180 0 1] )
%
% % Plot Cross-correlation - heavy - z
% subplot( 3, 2, 6 )
% for i = 1:n:imax
%
%     if ~isempty( acorrp{i,3} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'APA' )
%
%        plot( acorrp{i,3}(:,2), acorrp{i,3}(:,1), 'r.-' )
%        hold on
%
%     end
%
%     if ~isempty( acorrp{i,3} ) && strcmp( data{i,4}, 'Heavy' ) && strcmp( data{i,3},
'MEM' ) && strcmp( data{i,2}, 'TDP' )
%
%        plot( acorrp{i,3}(:,2), acorrp{i,3}(:,1), 'b.-' )
%        hold on
%
%     end
%
% end
% xlabel( 'Lag Time, l (s)' )
% ylabel( 'R_{x_{1}x_{100}}' )
% title( {'Z-Position Cross-correlation'; 'Heavy Aircraft, MEM'} )
% grid on
% axis( [0 180 0 1] )
```